



Proyecto Fin de Máster en
Ingeniería de Computadores

Curso 2009-2010

Análisis de aplicaciones interactivas en infraestructuras cloud

Carlos Martín Sánchez

Dirigido por:

Dr. Rubén Santiago Montero

Dr. Rafael Moreno Vozmediano

Dpto. Arquitectura de Computadores y Automática

Máster en Investigación en Informática,
Facultad de Informática, Universidad Complutense de Madrid

Autorización

El abajo firmante, matriculado en el Máster en Investigación en Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: "Análisis de aplicaciones interactivas en infraestructuras cloud", realizado durante el curso académico 2009-2010 bajo la dirección de los doctores Rubén Santiago Montero y Rafael Moreno Vozmediano en el Departamento de Arquitectura de Computadores y Automática (DACYA), y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en Internet y garantizar su preservación y acceso a largo plazo.

Firmado:
Carlos Martín Sánchez.

Resumen

En este trabajo se estudia la posibilidad de aplicar el paradigma de computación cloud computing para virtualizar aplicaciones interactivas. La motivación principal es poder estudiar la viabilidad de mover a una infraestructura cloud los puestos de prácticas de laboratorio de un entorno universitario.

Para poder realizar un estudio completo y riguroso de los retos que supondría esta adaptación de la facultad al cloud computing, en esta memoria se describe el diseño de una arquitectura para desktop cloud y su implementación como infraestructura prototipo funcional. Esta infraestructura es capaz de virtualizar y gestionar entornos completos de escritorio.

El uso de cloud computing para virtualizar entornos de escritorio interactivos es un área novedosa, en la que todavía no se disponen de mediciones de referencia para poder comparar el rendimiento de diferentes configuraciones y poder garantizar la calidad de servicio ofrecida. En esta memoria se proponen dos métricas de rendimiento especialmente diseñadas para adaptarse a las necesidades de la virtualización de puestos de laboratorios prácticos

Palabras clave

computación en la nube, nubes de escritorios, OpenNebula, virtualización, VNC, benchmark, laboratorio virtual, máquina virtual

Abstract

This work introduces a proposal to use the cloud computing paradigm to virtualize interactive applications. Its main goal is to study the feasibility to migrate the current university computer laboratories to a cloud environment.

In order to be able to perform a complete and accurate study of the challenges this proposal will have to face, this project describes the design of a desktop cloud architecture and its implementation as a working prototype infrastructure. It can virtualize and manage complete desktop environments.

The application of cloud computing in desktop virtualization is a novelty research area, in which there are not many benchmarks to a) compare how well different configurations perform, and b) guarantee the quality of service provided. In this project, two performance metrics are proposed. These metrics are designed specifically for the needs of a cloud infrastructure oriented to provide interactive desktop virtualization.

Key words

cloud computing, desktop cloud, OpenNebula, virtualization, VNC, benchmark, virtualized laboratories, virtual machine

Índice general

1. Introducción	1
1.1. Objetivos	3
2. Desktop & Campus Clouds	5
2.1. Conceptos previos	5
2.1.1. Virtualización	5
2.1.2. Cloud computing	8
2.2. Utilidad práctica	10
2.3. Casos de uso	11
3. Arquitectura para un Desktop Cloud	13
3.1. Front-end	16
3.2. Hosts	18
3.3. Almacenamiento	18
3.4. Conectividad de red	19
3.5. Máquinas Virtuales	19
4. Evaluación del Rendimiento	21
4.1. Medición del tiempo de despliegue	23
4.2. Propuestas para reducir el tiempo de despliegue	25
4.2.1. Rendimiento de los drivers ssh	26
4.2.2. Drivers de transferencia con cache de imágenes	28
4.2.3. Nuevos drivers con pre-copia	34
4.2.4. Conclusiones	40
4.3. Medición de la latencia de uso	40
4.3.1. VNCPlay	41
4.3.2. Rendimiento de la instalación	43
5. Conclusiones y Trabajo Futuro	46
Bibliografía	47

Capítulo 1

Introducción

Las actuales enseñanzas en las titulaciones de tecnologías de la información y las comunicaciones (TIC) requieren de entornos de entrenamiento que reproduzcan con la mayor fiabilidad posible las circunstancias que afrontará el alumno en su futura vida profesional. Esta práctica es común en otras enseñanzas, véase por ejemplo las titulaciones Bio-sanitarias que habitualmente incluyen sesiones prácticas con pacientes reales. Desgraciadamente este grado de experimentación no está al alcance de los alumnos de las titulaciones informáticas; y así, la formación que adquieren los estudiantes presenta carencias importantes en este área.

Probablemente la causa responsable de esta grave limitación es la necesidad de otorgar al alumno el control total de un entorno informático completo, tal y como lo hará en su futura práctica profesional. La enseñanza en un laboratorio de esta características conlleva ciertos problemas, a saber:

Seguridad.

El disponer del control total del entorno informático puede ocasionar un riesgo grave para las instalaciones de los equipos del laboratorio, o peor aún, para el resto de los equipos y usuarios de la misma red. Es importante destacar que el uso responsable o la correcta supervisión no siempre son suficientes para evitar este riesgo, ya que habitualmente los daños no se producen de forma intencionada.

Complejidad del Entorno.

Los laboratorios informáticos pueden requerir ciertas configuraciones complejas que deben estar listas antes de cada sesión práctica. A pesar de poder utilizar herramientas de automatización, normalmente se requiere la intervención del profesor o el personal técnico del laboratorio para terminar de ajustar el entorno de prácticas. Además el tiempo de instalación

y ajuste hace totalmente incompatible la creación o reparación del laboratorio con los horarios de los diferentes grupos de prácticas. La complejidad del entorno también hace que éste sea difícilmente reproducible por los alumnos en sus propios equipos.

El resultado es que el alumno realiza su aprendizaje en entornos limitados y aislados que no puede modificar; y desgraciadamente algunos aspectos sólo pueden estudiarse desde una perspectiva teórica. A modo de ejemplo, basta decir que actualmente los alumnos no pueden practicar la instalación de un Sistema Operativo (o ninguna otra aplicación), ni configurar su conexión de red.

Además, el protocolo de prácticas actual configura la organización de los laboratorios que resulta claramente ineficiente y poco flexible:

- Los alumnos, a pesar de disponer sus propios equipos, sólo pueden estudiar (prácticas no regladas) en las instalaciones de la Facultad.
- No todos los laboratorios disponen de todas las configuraciones, lo que hace que algunos laboratorios estén libres y otros sobre-demandados en los turnos de práctica libre; especialmente en los periodos de exámenes.

Finalmente, la situación descrita anteriormente no es singular en la Universidad Complutense, y es típica en las titulaciones informáticas de otras universidades [1, 2, 3, 4, 5].

Recientemente, el desarrollo de las técnicas de virtualización constituyen una prometedora tecnología que permitirá abordar los problemas detallados anteriormente. En general, la virtualización en informática hace referencia al conjunto de técnicas que emulan el comportamiento de un sistema o de parte de él. Así, una capa de software especializado (hipervisor, véase página 6) permite convivir simultáneamente múltiples sistemas operativos en ejecución (máquina virtual) usando la misma máquina física. Estas técnicas de virtualización ofrecen múltiples beneficios, como por ejemplo consolidación de servidores, gestión energética eficiente (Green IT) o despliegue sencillo de nuevas máquinas, entre otros. Además, es posible contratar la provisión de estas máquinas virtuales a proveedores externos (Cloud Computing) lo que abre una nueva vía de posibilidades para la gestión de infraestructuras informáticas [6].

En este proyecto se propone el uso de estas nuevas técnicas (virtualización y cloud) para aprovisionar de forma eficiente los entornos hardware y software necesarios para prácticas de laboratorios de informática, de forma que el alumno pueda experimentar con ellos sin ninguna restricción y de forma totalmente flexible (incluso desde su propia casa) [2, 3].

1.1. Objetivos

El objetivo principal de este proyecto consiste en explorar la aplicación de novedosas técnicas de virtualización y computación en Cloud a la gestión de los laboratorios universitarios de informática; de esta forma los alumnos podrían disponer de entornos de aprendizaje reales y complejos a los que poder acceder en cualquier momento y desde cualquier sitio. Estas nuevas técnicas permitirían además optimizar la gestión de las infraestructuras informáticas de las Facultades mediante el ahorro en coste de equipación, el aumento de la eficiencia energética y la flexibilización del servicio ofrecido mediante el uso de proveedores externos.

En general, el proyecto pretende integrar tecnologías de amplia aplicación en el campo de la virtualización y computación Cloud y centrarse en evaluar su viabilidad para el despliegue de laboratorios virtuales.

El proyecto se articula en torno a dos objetivos concretos:

Diseño de una infraestructura cloud piloto.

Se identificarán los requisitos necesarios de una instalación de este tipo. Se diseñará e implementará una infraestructura piloto que sea capaz de albergar las configuraciones típicas de las asignaturas que actualmente se cursan en las titulaciones de la Facultad de Informática. El sistema deberá ser lo suficientemente flexible como para virtualizar un amplio abanico de alternativas, tanto en complejidad (demanda de recursos); como en la naturaleza del entorno (basados en Windows y Linux).

Evaluación de las prestaciones y rendimiento de la infraestructura piloto.

Con el fin de validar el sistema desarrollado en el proyecto se realizará una evaluación doble. En primer lugar se estudiará el rendimiento desde un punto de vista puramente informático, estimando el comportamiento en situaciones de carga de trabajo elevada que sirvan como base para guiar un posible despliegue a gran escala. Por otro lado, se realizarán algunas sesiones gráficas “simuladas” en los entornos de laboratorios virtuales desplegados para poder comprobar la comodidad de uso del sistema desde el punto de vista del usuario final (los alumnos).

El diseño del sistema considerará como requisito el satisfacer la funcionalidad ofrecida hoy en día por un amplio número de asignaturas prácticas de las titulaciones impartidas en la Facultad de Informática. En particular el sistema de Laboratorios Virtuales deberá:

- Poder especificar diferentes tipos de puestos de prácticas virtuales, implementados como máquinas virtuales.

- Solicitar laboratorios completos como conjuntos de puestos virtuales.
- Desplegar y controlar estos puestos virtuales en una infraestructura de servidores centralizada.
- Deplegar y controlar puestos virtuales en proveedores externos.

La aportación de este trabajo es un estudio planteado como primer paso hacia la creación futura de un sistema más completo. El sistema que se espera obtener a largo plazo será capaz de automatizar y gestionar laboratorios virtuales mediante una sencilla interfaz gráfica web, que permita a los profesores configurar los entornos de prácticas. El alumno se podrá conectar por Internet a su puesto desde cualquier ordenador.

Capítulo 2

Desktop & Campus Clouds

Aunque no existen todavía un número elevado de artículos que permitan hablar de un amplio consenso, sí podemos decir que Desktop Cloud es el nombre que comúnmente reciben las infraestructuras cloud dedicadas a virtualizar entornos de escritorio [7]. Así mismo, Campus Cloud es como se denomina a los clouds dedicados a las necesidades específicas de los campus universitarios [8, 9].

En este capítulo se hace una breve introducción a los conceptos previos necesarios para poner en contexto la aportación principal de la memoria. Además se explican las ventajas del Cloud Computing en general, y en particular las de su aplicación en la Facultad.

2.1. Conceptos previos

2.1.1. Virtualización

En informática, la palabra virtualización se usa para referirse a una abstracción de recursos de computación, como la memoria, red, procesador, aplicaciones, etc. En este trabajo cuando se habla de virtualización, nos referimos a la virtualización de plataformas, que consiste en la abstracción del hardware para virtualizar un ordenador o sistema operativo en una máquina virtual (virtual machine, VM).

La virtualización tiene una larga historia, que comienza en los entornos de “mainframe” en la década de 1960, surgiendo como necesidad de ofrecer aislamiento entre usuarios. En ese momento se propuso la virtualización de sistemas operativos [10], y en la década de 1970 era ya una tecnología ampliamente usada [11].

El reciente interés que ha vuelto a poner de moda esta tecnología se originó al tener un único tipo de servidores caros y necesitar de algún modo compartir los recursos entre varias aplicaciones. Esto propició la idea de agrupar procesos que pertenecen al mismo usuario en su propio entorno virtual.

La tecnología de virtualización para arquitecturas i386 recientemente desarrollada ofrece nuevas formas de utilizar la tecnología informática al permitir acceso de forma lógica, en lugar de física, a datos, capacidad de cómputo, almacenamiento [12] y otros recursos.

Numerosas organizaciones están adoptando la tecnología de virtualización como medio para ofrecer servicios elásticos, agrupar la carga de servidores, y mejorar el ratio de utilización de los servidores. De esta manera se reducen los costes de hardware y mantenimiento, lo que conlleva además un recorte del consumo de energía, necesidad de refrigeración, y contaminación.

Hipervisor

El componente que administra o gestiona la virtualización se llama hipervisor (hypervisor), o también monitor de máquinas virtuales (virtual machine monitor, VMM). Se trata de una capa de software o firmware ejecutada sobre la máquina física, y presenta una abstracción del hardware físico a las máquinas virtuales que hospeda. El ordenador físico que ejecuta el hipervisor se conoce como anfitrión (host), y la máquina virtualizada como huésped (guest).

La máquina virtual puede simular todos los componentes hardware, como la memoria, disco, CPU, etc. Un hipervisor puede simular varias máquinas virtuales al mismo tiempo, y cada una de ellas puede contener diferentes sistemas operativos y ejecutar aplicaciones de manera independiente y simultánea.

El hipervisor permite manejar el ciclo de vida de las máquinas virtuales, de forma similar a las máquinas físicas: éstas pueden ser encendidas y apagadas, así como suspendidas y reanudadas. Además de eso, el hipervisor puede también hacer una copia del estado de la memoria de la máquina (snapshot), y restaurar la máquina cuando se desee en el mismo estado en el que se encontraba cuando se tomó la copia.

Esta tecnología pone también a nuestra disposición gestiones más complejas como la posibilidad de “migrar en vivo” (live migration), que consiste en “mover” la ejecución de la máquina virtual de una máquina física a otra de manera instantánea, sin necesidad de apagarla o detener las comunicaciones de red que pueda tener en curso.

Tipos de virtualización

En [13] resumen los diferentes tipos de virtualización agrupándolos en categorías según el grado de virtualización y el modo de funcionamiento del hipervisor.

Basándonos en el modo de funcionamiento, podemos distinguir hipervisores “non-hosted” y “hosted”. Mientras que los primeros funcionan directamente sobre el hardware (como por ejemplo Xen o ESX Server), los segundos se ejecutan sobre otro sistema operativo (como Microsoft Virtual PC [14] y Vmware Workstation[15]). La clasificación según el grado de virtualización es un poco más complicada. Las principales categorías son las siguientes:

Virtualización completa.

Implementa la plataforma completamente y permite el uso de un sistema operativo virtual. El SO invitado está totalmente abstraído de los elementos físicos y no requiere ninguna modificación para ser virtualizado. Algunos ejemplos de virtualización completa son Qemu [16], Vmware ESX [17], Vmware Workstation [15], Virtual Box [18], etc.

Virtualización asistida por hardware.

La virtualización que es parcialmente realizada por hardware. Algunas de las instrucciones del SO tienen acceso directo al hardware, sin necesidad de ninguna modificación o emulación en el SO. Ejemplos de esta tecnología son IBM POWER, las instrucciones Intel VT y AMD V. En Linux, KVM [19] es el sistema virtualizador asistido por hardware más avanzado.

Virtualización parcial.

Consiste en la abstracción de algunos componentes específicos. Esta solución permite la división de recursos y aislar los procesos, pero no arrancar otro sistema operativo. Por cada proceso y usuario se usa un espacio de direcciones diferente.

Para-virtualización.

Permite a un ordenador ejecutar varios sistemas operativos. Estos SSOO tienen que ser modificados para entender llamadas del huésped. Esto significa modificar el núcleo del SO para reemplazar instrucciones no virtualizables con llamadas al hipervisor. Un ejemplo es Xen [20]. Los sistemas para-virtualizados sufren limitaciones para combinar diferentes sistemas, migraciones de huéspedes de una plataforma a otra, y dependencias del kernel, pero tienen mucho mejor rendimiento que las virtualizaciones parciales o totales por su dependencia de la carga de trabajo.

La virtualización del SO únicamente.

Sin virtualización del hardware, lo que significa que varias instancias del

mismo SO pueden ejecutarse en el mismo hardware. Ejemplos de este tipo de virtualización son OpenVZ y LinuxVserver.

2.1.2. Cloud computing

Cloud computing, traducido como computación en la nube, es un paradigma de computación basado en internet, en el cual los recursos son proporcionados bajo demanda. Comenzó su popularidad en 2006 con el lanzamiento de Amazon EC2 [6], que ofrecía máquinas virtuales bajo demanda, cobrando por hora de uso. Este modelo de contratación de recursos se conoce como “infraestructura como servicio” (infrastructure-as-a-service, IaaS), y está asociado a lo que comúnmente se conoce como cloud computing.

Algunas de las ventajas destacables del paradigma de computación cloud computing son las siguientes:

Costes.

Muchas aplicaciones empresariales requieren una gran infraestructura para funcionar. Esas infraestructuras grandes no son baratas, y además conllevan costes asociados como la refrigeración de las instalaciones.

Con el cloud computing las organizaciones pueden prescindir de mantener su propia infraestructura, y contratarla como servicio a un proveedor externo. Esto ahorra costes al pagar solo por la infraestructura, almacenamiento y red realmente usados, además de eliminar todos los costes de mantenimiento de la infraestructura local.

Escalabilidad.

La escalabilidad es una cualidad deseada desde el comienzo de la informática. Esta cualidad cobra más relevancia en el caso de aplicaciones accesibles por internet, en las cuales el número de usuarios del sistema es difícil de predecir.

Las aplicaciones diseñadas para ser ejecutadas en el cloud pueden expandirse automáticamente según la carga de trabajo y la demanda, así como contraerse para consumir los recursos óptimos en todo momento.

Más fiables

Las aplicaciones que se desarrollan en el cloud tienen la ventaja de que pueden estar replicadas en varios “datacenters” en lugares diferentes. En el caso de un fallo de hardware, las aplicaciones pueden migrar de localización o simplemente expandirse de nuevo.

Sostenibilidad medioambiental

Por el diseño propio del paradigma, cloud computing ofrece una mayor

eficiencia y un uso más responsable de los recursos. La habilidad de desplegar y finalizar instancias de una aplicación según la demanda de uso inmediata es mucho más eficiente que ejecutar esa misma aplicación en su propio servidor dedicado. Además, la infraestructura ofrecida por un proveedor cloud agrupa en un mismo lugar los recursos físicos, reduciendo las necesidades de refrigeración, consumo de potencia, etc.

Gestor de Infraestructuras Virtuales

Los beneficios del paradigma Cloud Computing pueden aplicarse también en infraestructuras privadas. Con un cloud privado, los usuarios de la compañía u organización pueden aprovechar todas las ventajas del cloud computing sin tener que depender de compañías proveedoras externas.

El componente software encargado de convertir una infraestructura capaz de virtualizar en una infraestructura cloud se conoce como gestor de infraestructuras virtuales. Los requisitos que debe cumplir un gestor de infraestructuras cloud son los siguientes:

- Representar los recursos disponibles de manera homogénea, independientemente de la plataforma de virtualización usada (hipervisor, pág. 6).
- Gestionar el ciclo de vida de las máquinas virtuales completamente. Esto incluye la creación de redes para interconectar grupos de máquinas virtuales, y los requisitos de almacenamiento como la gestión de imágenes y la creación de nuevos discos virtuales dinámicamente.
- Ser capaz de distribuir las máquinas virtuales de forma que configurable, dando la posibilidad de seguir políticas configurables.
- Adaptación a una demanda de recursos variable, incluyendo picos de carga en los que la infraestructura local sea insuficiente. Debe ser capaz también de adaptarse a cambios en la infraestructura virtualizadora

OpenNebula es un gestor de infraestructuras virtuales que permite crear clouds privados, públicos e híbridos. Cualquier organización puede usarlo para desplegar y gestionar máquinas virtuales, en recursos locales o clouds públicos externos. Se encarga de automatizar la preparación de la máquina virtual (gestionando las imágenes de discos, configurando las redes, etc.) sin importar qué capa de virtualización se usará como hipervisor (KVM, Xen, VMware) o si la máquina virtual será desplegada en un proveedor externo, como EC2.

2.2. Utilidad práctica

La aplicación de una infraestructura cloud para virtualizar laboratorios de prácticas resulta de interés en dos aspectos: la experiencia del alumno, y el impacto en la Facultad. Desde el punto de vista del alumno el proyecto presenta una utilidad práctica clara en cuatro áreas:

Entornos reales de práctica.

Gracias a los puestos de laboratorio virtuales el alumno puede practicar sin ninguna restricción en entornos similares a los que se enfrentará en su actividad profesional. Esto permitirá la realización de prácticas avanzadas que hoy en día están restringidas por razones de seguridad o por el coste que supondría re-instalar los laboratorios después de cada sesión.

Entornos de práctica complejos.

Además, los puestos virtuales permitirán que los alumnos se familiaricen con disposiciones complejas que supongan la intervención de más de un puesto de laboratorio. Actualmente un alumno sólo dispone de un ordenador para realizar su trabajo; en informática el uso de sistemas distribuidos y redes de computadores es un área esencial que necesariamente requiere de varios ordenadores. Gracias al proyecto el alumno podrá disponer de varios puestos simultáneamente para controlar su propia red virtual de computadores o granja.

Entornos de práctica seguros.

La virtualización garantiza la estanqueidad de los puestos de prácticas. Así, el alumno puede realizar su trabajo sin interferir con otros usuarios de la red. Además dado que los puestos virtuales se replican antes de cada uso, el alumno puede practicar con la propia configuración del puesto sin miedo a “romper” ningún ajuste, ya sea de forma intencionada o no.

Facilitar el acceso a los entornos de práctica.

Uno de los problemas más importantes que hay que afrontar en la gestión de los laboratorios Informáticos es la gran demanda de prácticas libres que solicitan los alumnos para preparar sus trabajos de clase. Esto supone disponer de puestos con las configuraciones adecuadas para cada asignatura tanto para el estudio como para las prácticas regladas. El resultado son laboratorios altamente demandados y alumnos que no encuentran puestos para practicar. Los puestos virtuales se pueden acceder desde cualquier ordenador, lo que solucionaría el problema ya que el alumno podría usar el laboratorio de la Facultad desde su casa o la sala de estudio mediante la conexión Wi-fi.

Además también hay potenciales beneficios para la Universidad, respecto a la gestión de su infraestructura informática:

Ahorro en el coste de la infraestructura informática.

Los puestos virtuales se pueden acceder desde cualquier equipo, así no es necesario comprar ordenadores de gama alta para los laboratorios. Además al centralizar los servidores que soportan los puestos virtuales es posible adaptar los laboratorios a las necesidades, algo totalmente imposible hoy en día. Finalmente la posibilidad de externalizar los puestos virtuales permite ajustar la infraestructura a su uso medio y usar proveedores externos en situaciones de demanda pico, como por ejemplo la época de exámenes.

Ahorro energético.

Al consolidar puestos informáticos el consumo energético necesario para mantener su operación se reduce considerablemente.

Gestión eficiente de los espacios de la facultad.

Es cada vez más común que los alumnos dispongan de su propio ordenador portátil, especialmente desde la introducción de los “netbook”. Gracias a la infraestructura de red inalámbrica de la Universidad Complutense y los puestos virtuales cualquier espacio puede convertirse en un laboratorio. Un profesor puede solicitar puestos virtuales para su uso durante la clase que va impartir de forma que los alumnos practiquen en su horario habitual, evitando la problemática que supone organizar sesiones prácticas no programadas al principio del curso.

2.3. Casos de uso

Para clarificar los objetivos anteriores y comprender las ventajas que la implantación de una infraestructura cloud traería a la Facultad, se incluyen a modo de ejemplo algunos casos de uso que se daría al sistema.

Solicitar un laboratorio para una sesión práctica no programada.

Un profesor de la Facultad de Informática decide realizar una sesión de laboratorio para practicar algunos de los conceptos explicados durante la última semana; se conecta al servicio de laboratorios de la facultad y realiza la solicitud de 15 puestos virtuales para la siguiente clase. Unos minutos antes de la clase se arrancan los puestos virtuales, que los alumnos pueden usar desde cualquier ordenador.

Adaptar los laboratorios al período de exámenes.

Los alumnos solicitan el puesto específico para la asignatura que quieren preparar. Los puestos virtuales se arrancan en las instalaciones según las demandas

de los alumnos sin límite en el número de puestos de un tipo determinado. Las peticiones para usar los laboratorios desbordan la capacidad de la Facultad debido a un fallo de los sistemas, y el Decanato de la Facultad decide contratar puestos virtuales a 0.1€/hora para que todos los alumnos puedan preparar los exámenes.

Cuando llega el periodo de exámenes ya no será necesario cerrar algunos laboratorios para la puesta a punto de los equipos de cara a los exámenes, de manera que los alumnos pueden preparar mejor las asignaturas. Además se evitan masificaciones en los laboratorios más demandados.

Estudiar en los laboratorios de la facultad desde casa.

Un alumno va a preparar una asignatura durante las vacaciones de Navidad, y para ello solicita mediante el interfaz un puesto de laboratorio Virtual a la facultad. El alumno puede preparar la asignatura desde su casa.

Capítulo 3

Arquitectura para un Desktop Cloud

Este capítulo detalla el diseño de una arquitectura hardware y software capaz de albergar soluciones cloud en el escenario de la Facultad. Tras decidir qué componentes deben formar la infraestructura, se ha implementado un prototipo funcional, capaz de virtualizar puestos de prácticas bajo demanda que pueden ser usados gráficamente mediante VNC.

La implementación de una arquitectura cloud debe cumplir con unos requisitos que garanticen una funcionalidad mínima, explicados en la sección de introducción al Cloud Computing, pág. 9. La implantación de un Desktop Cloud en la Facultad conlleva además los siguientes requisitos específicos (expuestos en la sección de Objetivos, pág. 3):

- Poder especificar diferentes tipos de puestos de prácticas virtuales, implementados como máquinas virtuales.
- Solicitar laboratorios completos como conjuntos de puestos virtuales.
- Desplegar y controlar estos puestos virtuales en una infraestructura de servidores centralizada.
- Desplegar y controlar puestos virtuales en proveedores externos.

La infraestructura prototipo diseñada se trata de un cloud privado. Un cloud privado es un sistema que permite utilizar servidores propios para virtualizar máquinas, con una gestión centralizada de imágenes, máquinas virtuales, usuarios, redes, hosts (nodos físicos), grupos de clusters, etc. El sistema expone esa capacidad a los empleados de la organización o, en este caso, a los profesores y alumnos de la facultad.

El sistema prototipo implementado no se trata de una simulación o descripción teórica, sino de un conjunto real de máquinas dedicadas a la realización de esta memoria. La arquitectura y los componentes usados son los mismos que se usarían en el entorno real de la facultad, solo que a menor escala y con capacidades más modestas.

Este prototipo no pretende ofrecer un servicio completo en el que poder realizar sesiones de laboratorios. La finalidad es disponer de un sistema que permite identificar los requisitos y necesidades propias de las aplicaciones interactivas que tendría que virtualizar la Facultad: puestos de los laboratorios actualmente usados en las titulaciones.

El diseño de la arquitectura se compone de dos tipos de máquinas: el front-end, que contiene el gestor de la infraestructura virtual; y los hosts, nodos encargados de virtualizar los puestos de laboratorio mediante un hipervisor.

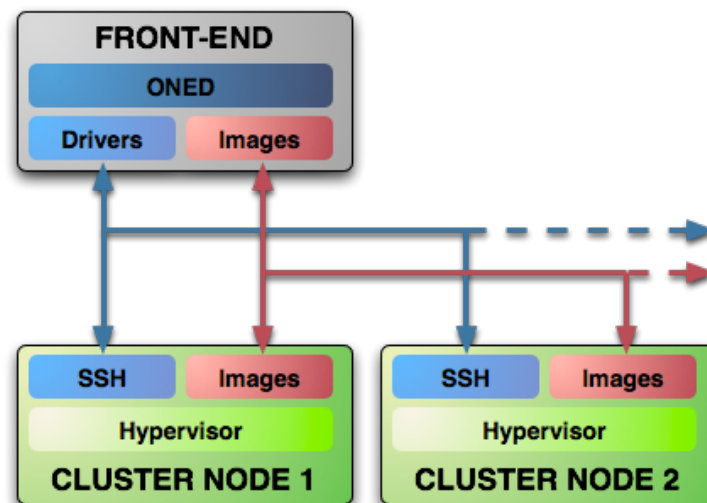


Figura 3.1: Arquitectura del sistema implementado.

A la hora de escoger los componentes que forman la infraestructura, se ha tenido como meta la siguiente lista de características:

Escalabilidad.

Aunque se trate de un prototipo, la infraestructura creada se ha diseñado de modo que pueda ser replicada en un entorno real, con una carga de trabajo como la de la Facultad.

Capacidad de desplegar un conjunto heterogéneo de máquinas virtuales.

La infraestructura no impone ninguna restricción en cuanto a qué sistema

operativo puede instalarse en las máquinas virtuales, más allá de la arquitectura hardware (i386 ó x86-64). Las máquinas virtuales que pueden crearse son además fácilmente adaptables a los requisitos de las diferentes asignaturas, tanto en su configuración hardware, como las conexiones de red entre grupos de máquinas.

Seguridad y aislamiento.

El sistema permite que un alumno tenga el control absoluto de su puesto de trabajo (o conjunto de máquinas virtuales). Las instancias de las máquinas virtuales están aisladas unas de otras, y las acciones de cada alumno no interfieren con los demás.

De este modo, un uso malintencionado o un error a la hora utilizar la máquina virtual queda solucionado con sólo apagar la máquina, sin consecuencias para las siguientes instancias que serán creadas desde el estado inicial.

Este aislamiento se aplica también a las redes. El gestor de infraestructuras virtuales OpenNebula permite crear cuantas redes virtuales sean necesarias, de forma que una sesión práctica que requiera el uso de varias instancias de máquinas virtuales interconectadas se realizará sin colisiones ni interferencias con el resto de alumnos.

Solicitar laboratorios completos como conjuntos de puestos virtuales.

El sistema responde a las peticiones simultáneas de un número elevado de máquinas virtuales de la misma asignatura. Esto es indispensable, teniendo en cuenta el modo de usar los laboratorios.

Gestión inteligente de los recursos físicos.

La infraestructura es capaz de controlar un número variable de hosts, sin requerir que sean homogéneos. Esto permite la ampliación de la infraestructura física cuando se considera necesario, pudiendo incluir a la granja de servidores ordenadores nuevos, con configuraciones hardware diferentes e incluso con hipervisores de virtualización diferentes.

Esto supone una ventaja clara, ya que actualizar la infraestructura atendiendo a una demanda creciente no implica tener que revisar toda la arquitectura por la compra de dos servidores y sin romper compatibilidad con herramientas creadas sobre este prototipo.

Desplegar y controlar puestos virtuales en proveedores externos.

OpenNebula permite acceder a máquinas virtuales en proveedores cloud externos, como Amazon EC2 o ElasticHosts. Esto significa que aunque el prototipo no se haya usado como cloud híbrido, sí que tiene la posibilidad de usar esta funcionalidad si llega a implantarse de forma definitiva. Esto permite contratar capacidad extra de manera puntual a proveedores externos en periodos breves de alta demanda que no justifiquen la compra de nuevo hardware.

A continuación se detallan brevemente las características principales del diseño de la infraestructura, y los recursos físicos dedicados al prototipo.

3.1. Front-end

El front-end ejecuta el Virtual Infrastructure Manager, gestor de infraestructura virtual. El gestor de infraestructura virtual, o framework de cloud escogido ha sido OpenNebula. Las características que hacen destacar este framework sobre otras alternativas son las siguientes:

Es open source.

Esto garantiza que puede ser modificado en un futuro para dar soporte a nuevos requisitos de la Facultad, y afrontar retos que quizás hoy no son conocidos.

Gran escalabilidad.

Es importante disponer de la garantía de que el sistema prototipo puede ser usado a gran escala sin tener que cambiar los componentes clave, tan solo la envergadura de la infraestructura.

Estabilidad.

Otra de las grandes virtudes que hacen de OpenNebula una elección idónea, es que no sólo presenta avanzados logros tecnológicos con los que hacer pequeñas infraestructuras de prueba, sino que es capaz de proporcionar un sistema robusto en el que se puede confiar.

Su arquitectura es modular y extensible.

OpenNebula controla los hosts mediante un sistema de plugins. Es muy sencillo modificar estos plugins o incluso crear nuevos para ajustar perfectamente el rendimiento a la infraestructura física disponible. Además, expone su funcionalidad mediante varias interfaces, lo que facilita la creación de herramientas de alto nivel sin imponer un modo único de uso.

La figura 3.2 muestra de manera simplificada la arquitectura interna de OpenNebula. Los componentes principales son el OpenNebula Daemon o demonio del sistema, el Scheduler o planificador, y los diferentes drivers.

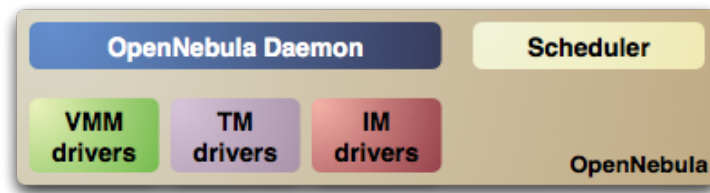


Figura 3.2: Arquitectura interna de OpenNebula.

Estos son los tres tipos de procesos existentes:

OpenNebula daemon

El demonio OpenNebula es el núcleo de la arquitectura. Se encarga de coordinar las operaciones del resto de módulos, y controla el ciclo de vida de las máquinas virtuales.

Drivers

OpenNebula sigue un diseño modular, y las operaciones y comunicaciones con los hosts se realizan mediante drivers intercambiables. Esto permite usar un driver diferente para cada host (por ejemplo, si cada uno tiene instalado un hipervisor diferente), y también modificarlos y crear drivers nuevos sin comprometer la estabilidad de OpenNebula.

A continuación se explica cada tipo de driver:

VMM, Virtual Machine Manager: Controlan el ciclo de vida de las máquinas virtuales en el hipervisor.

TM, Transfer Manager: Gestionan la transferencia de archivos entre el front-end y los hosts.

IM, Information Manager : Monitorizan el estado de los hipervisores, y de las máquinas virtuales que éstos ejecutan.

Scheduler

El planificador en la arquitectura de OpenNebula es un componente totalmente desacoplado del núcleo, y puede ser reemplazado por cualquier otro software que use la misma interfaz. Este componente se encarga de distribuir las máquinas virtuales en los hosts disponibles. El planificador incluido con la instalación de OpenNebula permite usar políticas de distribución, y también un sistema de prioridades. Haciendo uso de estas facilidades, la infraestructura local puede ser usada de forma totalmente personalizada a las necesidades propias.

OpenNebula no requiere un hardware potente, y las dependencias software para poder ser ejecutado se cumplen en cualquier distribución de GNU/Linux

actual. Ya que el rendimiento del sistema front-end no precisa de una máquina dedicada, OpenNebula se ha instalado sobre Ubuntu server 9.04 en una máquina virtual.

La máquina virtual dispone de 512 MB de RAM y una CPU virtual, más que suficiente para el sistema operativo y OpenNebula. El sistema operativo está instalado en un disco duro virtual de 5 GB, y dispone además de otro disco con 15 GB usado como repositorio de imágenes registradas.

En cuanto a los interfaces de red, la máquina tiene un interfaz público para poder conectarse por ssh, y otro privado con el que se comunica con los cuatro hosts.

3.2. Hosts

Los hosts son máquinas físicas dedicadas a virtualizar y controlar el ciclo de vida de las máquinas virtuales gestionadas por OpenNebula.

La infraestructura prototipo consta de cuatro máquinas físicas de formato rack. Cada una de ellas tiene una CPU de 8 núcleos Intel(R) Xeon(R) L5335 @ 2.00GHz y 8 GB de RAM. El sistema operativo instalado se trata de la distribución GNU/Linux Ubuntu 10.04 server.

El único componente software necesario para poder usar estos equipos como hosts es el hipervisor. Como se ha explicado anteriormente, existen multitud de alternativas, muchas de ellas soportadas por OpenNebula. La elección ha sido KVM, por los siguientes motivos:

- Es open source
- Forma parte del núcleo de Linux, por lo que aprovecha su gestión de memoria y planificador de actividad. Esto hace de KVM muy ligero, imponiendo una sobrecarga mínima.
- Hace uso de las extensiones de hardware para la asistencia de virtualización Intel VT y AMD V.

3.3. Almacenamiento

En cuanto al almacenamiento, cada host dispone de un disco duro de 80 GB dedicado exclusivamente a las imágenes de las máquinas virtuales; mas un almacenamiento de 370 GB compartido por NFS. Desgraciadamente, el almacenamiento compartido no pudo usarse de manera exclusiva para realizar los

experimentos de este proyecto, por lo tanto las medidas de rendimiento de los drivers NFS daban unos resultados poco fiables, de un rendimiento muy inferior al esperado. Por ese motivo, se prescindió de su uso, y en esta memoria las mejoras relativas al almacenamiento se han centrado en el almacenamiento no compartido.

3.4. Conectividad de red

Los cuatro hosts tienen dos interfaces de red cada uno. Uno de ellos tiene salida a internet, y el otro está conectado a un switch privado. La gestión y monitorización que hace OpenNebula se lleva a cabo mediante conexiones ssh a través de la red privada.

3.5. Máquinas Virtuales

Las máquinas virtuales en OpenNebula se crean mediante plantillas, que de manera simplificada constan de los siguientes elementos:

- Capacidad, de CPU y memoria.
- Un conjunto de interfaces de red, asociados a redes virtuales también gestionadas por OpenNebula.

Las máquinas virtuales pueden ser creadas con tantas interfaces de red virtuales como se necesiten (incluso con ninguna). Estos interfaces pueden tener acceso tanto a una red privada virtual gestionada por OpenNebula, como a internet.

- Un conjunto de imágenes de discos duros. Estos discos pueden usar alguna de las imágenes registradas en el catálogo, o una imagen vacía creada únicamente para la cada una de las instancias desplegadas a partir de la plantilla.

Las plantillas de las máquinas virtuales permiten también contextualizarlas. Mediante este mecanismo, es posible crear diferentes máquinas virtuales a partir de una única imagen, pero con varios ficheros de configuración que permiten personalizar cada instancia. Se expone a continuación un ejemplo de la utilidad de la contextualización en el contexto de la Facultad:

El profesor genera una imagen con el sistema operativo y los programas necesarios para las prácticas de su asignatura, y la registra en el sistema. Al

lanzar una instancia de esta imagen, OpenNebula puede contextualizarla de modo que el sistema operativo tenga un usuario accesible únicamente mediante la clave pública del alumno que ha solicitado la creación de la instancia.

Capítulo 4

Evaluación del Rendimiento

En este capítulo se discute cómo medir el rendimiento de una infraestructura cloud orientada a ofrecer virtualización de escritorio. Para ello, y siguiendo con el escenario de los laboratorios de la Facultad como referencia, se definen a continuación los diferentes perfiles de gente que tendrá que interactuar con el sistema, y qué esperará cada uno de él para considerarlo satisfactorio.

Administradores de la infraestructura cloud.

El personal de administración vería su tarea facilitada si los laboratorios son virtualizados. La principal ventaja sería pasar de administrar una cantidad elevada de máquinas individuales, con configuraciones diferentes para cada asignatura y dispersas en los laboratorios, a tener que ocuparse de un conjunto reducido de máquinas (los hosts y el gestor de infraestructura OpenNebula).

Para este perfil, la implantación de la infraestructura cloud sería satisfactoria si la administración y mantenimiento no se vuelve más compleja. Cuantificar esto con un prototipo a pequeña escala es complicado, por lo que habría que implantar el cloud y esperar a conocer el número de incidencias generadas tras unos meses de uso intenso. Por ello, la infraestructura prototipo no puede ayudar en este aspecto.

Profesores.

Con la ayuda de la infraestructura cloud, pueden preparar prácticas mucho más complejas y variadas, al dar a los alumnos control total de las máquinas. Además pueden pensar prácticas en las que el alumno controle varias máquinas, útil para enseñar conceptos como redes, tecnologías web, programación distribuida, etc.

Para que los profesores consideren el rendimiento de la infraestructura un éxito, se tendría que cumplir que estas ventajas no sean empañadas

por inconvenientes en el modo de realizar las prácticas, que supongan un detrimento en la facilidad de uso para el alumno.

Alumno

Para el alumno, una infraestructura cloud supondría tener la oportunidad de realizar prácticas que actualmente no le son ofrecidas. Además, contaría con la libertad de poder estudiar y practicar con las máquinas virtuales en cualquier horario, incluso cuando la facultad esté cerrada conectándose desde casa.

Los alumnos acogerían un nuevo sistema cloud si la experiencia de uso es equivalente o mejor a la de los puestos físicos actuales.

Las interesantes nuevas formas de diseñar y realizar prácticas de laboratorio en la infraestructura cloud no sirven de nada si no se puede garantizar un buen desempeño desde el punto de vista del usuario final: los alumnos. Queda claro por tanto, que antes de implantar una infraestructura a escala real son necesarias herramientas que ayuden a razonar y cuantificar cómo de bueno es el sistema para los usuarios finales, no sólo desde el punto de vista tecnológico.

El rendimiento en cloud computing normalmente se mide por la cantidad de tareas realizadas, o la capacidad de procesamiento. En el caso de la virtualización de escritorios, y el de la facultad en particular, la capacidad de cómputo no garantiza que los requisitos explicados arriba se cumplan [21, 22, 23].

Por lo tanto, se ha establecido que las métricas que definen cómo de bueno es el sistema propuesto en el caso específico de la virtualización de laboratorios de prácticas son principalmente las dos siguientes:

Tiempo de despliegue

En este proyecto se define el tiempo de despliegue como el tiempo que pasa desde que se solicita una máquina virtual, hasta que ésta está totalmente iniciada y lista para ser usada por una conexión remota. La máquina se considera totalmente iniciada cuando el sistema operativo ha terminado de iniciarse, incluyendo el entorno gráfico.

Actualmente no existen precedentes en la literatura que consideren como medida de rendimiento el tiempo que tarda una infraestructura cloud en desplegar las máquinas virtuales. En el caso de la Facultad, hacer los laboratorios virtualizados igual de cómodos de usar que los actuales puestos físicos es un requisito; lo que significa que iniciar un puesto de prácticas deberá tardar un tiempo aceptable.

Desplegar una máquina virtual en la infraestructura cloud puede conseguirse fácilmente entre 2 y 5 minutos, dependiendo de la configuración del sistema.

En el escenario de la Facultad, cuando una clase práctica comience se pedirán un número elevado de puestos de la misma asignatura simultáneamente. Para evitar tener un laboratorio lleno de alumnos esperando a que sus puestos virtuales se inicien, la infraestructura debe contar con un modo de medir el tiempo de respuesta a peticiones elevadas simultáneas. Esto permitirá validar la infraestructura prototipo implementada en este proyecto, y optimizar instalaciones de mayor envergadura posteriormente.

Latencia de uso

La latencia de uso es la lentitud del sistema que el usuario del escritorio virtualizado percibe. Otra forma de definirlo sería cómo de ágil es la respuesta del escritorio a las acciones interactivas del usuario. Un ejemplo más concreto de esta latencia de uso es la medición del tiempo que pasa desde que el usuario hace click en un botón, hasta que aparece el menú desplegable correspondiente.

La mayoría de benchmarks disponibles para virtualización en infraestructuras cloud no están pensadas desde el punto de vista de un usuario que se conecta remotamente al escritorio virtualizado. En su lugar, miden el rendimiento de ciertas aplicaciones, o el tiempo necesario para la ejecución de operaciones concretas [24].

En varios de los artículos que tratan sobre la virtualización de escritorios, se menciona el tiempo de respuesta a las acciones del usuario como un aspecto determinante del rendimiento del sistema estudiado [25, 26, 22, 27, 23].

Si la respuesta del escritorio virtualizado accedido remotamente es significativamente más lenta que la del mismo escritorio usado en un ordenador de sobremesa, la infraestructura cloud sería rechazada por los alumnos. Por ese motivo, esta métrica es relevante para poder estudiar la viabilidad de la infraestructura prototipo.

Aparte de la viabilidad del sistema, es importante contar con herramientas capaces de medir esta métrica de rendimiento. Así, a la hora de desarrollar una infraestructura cloud orientada a la Facultad se podrán comparar distintas configuraciones alternativas de manera rigurosa.

4.1. Medición del tiempo de despliegue

Como se ha explicado en el capítulo anterior, el tiempo de despliegue de una máquina virtual es el tiempo transcurrido desde que se solicita la misma al gestor del cloud OpenNebula, hasta que el sistema operativo virtualizado ha terminado de iniciarse, incluyendo el escritorio.

Los resultados presentados en este capítulo fueron realizados con una máquina virtual corriendo el sistema operativo Ubuntu 10.04. La imagen registrada para esta máquina virtual es de un tamaño de 4 GB.

Aparte de medir el tiempo total de despliegue, los scripts desarrollados muestran los tiempos parciales. Esto ayuda a identificar qué operaciones de la infraestructura cloud forman el cuello de botella, y requieren ser mejorados. Desde que la máquina virtual es solicitada a OpenNebula hasta que está lista para su uso podemos diferenciar las siguientes fases:

Deploy

Las máquinas virtuales solicitadas a OpenNebula no son creadas instantáneamente en el hipervisor de alguno de los hosts. En su lugar la máquina o, más bien, la descripción de la máquina que se quiere ejecutar, se crea en un estado pending (pendiente). La máquina permanece en ese estado hasta que el planificador ordena desplegarla en uno de los hipervisores. El tiempo transcurrido en esta fase está representado en color azul en las gráficas siguientes.

Boot

La siguiente fase por la que la máquina pasa en el gestor OpenNebula es el estado prolog. Durante este estado, el gestor cloud realiza operaciones preparativas para poder crear la máquina virtual solicitada en el hipervisor destino. El tiempo que dura esta fase debe principalmente a la copia de los ficheros de imágenes.

Una vez que los preparativos están terminados, la máquina pasa al estado boot (arranque).

Running

El siguiente paso en el ciclo de vida de la máquina virtual es comenzar la virtualización. Durante esta fase, representada en naranja, OpenNebula ordena al hipervisor correspondiente que cree una instancia de la máquina virtual. La máquina permanece en el estado boot, hasta que el hipervisor notifica que ha creado la instancia, y que la máquina está corriendo. En ese momento, el estado de la máquina en OpenNebula es running.

Lista para el usuario

El tiempo que tarda el sistema operativo virtualizado en inicializarse completamente, incluido el inicio de sesión en el escritorio Gnome, se corresponde al color verde.

Para recopilar los datos presentados en esta memoria, se desarrollaron unos scripts que leen los archivos de log creados por OpenNebula, en donde se puede

encontrar en qué momento se producen los cambios de estado. La única información que no es posible encontrar en los logs del front-end es el momento en el que el sistema operativo virtualizado termina de inicializarse.

La imagen del sistema operativo ha sido configurada para iniciar sesión en el escritorio Gnome de manera automática, sin requerir la contraseña del usuario. También está contextualizada para ejecutar un script, añadiéndolo en la sección de preferencias “Aplicaciones al inicio”. Este script guarda el momento en que es ejecutado en un pequeño fichero de texto, y lo copia mediante una transferencia ssh al front-end, para poder ser leído por los scripts mencionados.

4.2. Propuestas para reducir el tiempo de despliegue

En este capítulo se pretenden mostrar dos características clave de la infraestructura prototipo implementada, a saber:

Las mediciones propuestas sirven para localizar cuellos de botella.

Aparte de poder validar una instalación, las mediciones obtenidas permiten comparar diferentes alternativas de manera rigurosa, con información suficiente como para estudiar por separado el rendimiento del planificador, los drivers, el hipervisor, o el sistema operativo que se está virtualizando.

OpenNebula es flexible y se adapta a la infraestructura.

En este capítulo se proponen dos mejoras en los drivers de transferencia de imágenes ssh que cambian el concepto del repositorio de imágenes original: un directorio en el front-end desde el que se copian ficheros a los hosts.

Los cambios introducidos no requieren más que modificar scripts, de una complejidad media para cualquiera que esté familiarizado con su función. Estos cambios, que introducen una nueva forma de usar el repositorio de imágenes proporcionado por OpenNebula, no sólo son simples de realizar sino que encajan correctamente con el resto del sistema. La creación de máquinas virtuales no tiene que ser adaptada de ningún modo, por lo que cualquier herramienta de un nivel de abstracción mayor que use OpenNebula podrá beneficiarse inadvertidamente de las nuevas mejoras y el mejor rendimiento.

Como el objetivo de esta memoria no es conseguir una infraestructura lista para ser usada en producción a gran escala, se determinó que era más interesante explorar la capacidad de medir mejoras del rendimiento que el hecho de alcanzar ese rendimiento. Por lo tanto, los dos nuevos drivers presentados en este capítulo no aspiran a convertirse en una alternativa a las soluciones de almacenamiento compartido de alto rendimiento actuales.

4.2.1. Rendimiento de los drivers ssh

Como se ha explicado en la página 17, OpenNebula cuenta con tres tipos de drivers. Cada host que OpenNebula gestiona tiene asociado un driver de cada tipo.

Information Manager

Encargados de monitorizar el estado del hipervisor, y las máquinas que éste está virtualizando. En la infraestructura prototipo, se han usado los drivers para KVM.

Virtual Machine Manager

Drivers encargados de controlar el ciclo de vida de la máquina, operando sobre el hipervisor. En el caso del prototipo, los drivers para KVM fueron usados.

Transfer Manager

Estos drivers gestionan el movimiento de los ficheros de las imágenes.

Los drivers de transferencia de archivos (transfer manager drivers) ssh realizan la copia mediante el comando scp. La copia se realiza desde una carpeta local en el front-end, a un directorio del disco duro local del host en el que la máquina será desplegada. Estos drivers ssh están pensados para infraestructuras simples, sin almacenamiento compartido o en las que el tiempo de transferencia de imágenes no es relevante, bien porque las imágenes sean de tamaño reducido o porque su copia se realice de manera muy poco habitual.

Es de esperar que estos drivers no tengan un buen rendimiento en las mediciones del tiempo de despliegue. Para comprobar esta suposición, se desplegaron máquinas de manera simultánea, en grupos de 1 hasta 5. Entre el despliegue de cada grupo, las máquinas virtuales fueron apagadas, y eliminadas completamente del host, para comenzar las mediciones de cada grupo desde el mismo estado inicial. La figura 4.1 muestra el tiempo total que tardaron los grupos de máquinas en estar totalmente iniciadas. Puede apreciarse una relación clara entre el número de peticiones simultáneas y el tiempo necesario.

Como se ve en la gráfica, el tiempo de despliegue es bastante inaceptable, teniendo en cuenta que la referencia de lo que se considera como un tiempo normal es el encendido de un puesto de sobremesa actual. Esperar alrededor de 7 minutos para iniciar una máquina ya parece elevado, pero tener que esperar más de 30 minutos para tan sólo 5 puestos virtualizados es inadmisibles, ya que hace la infraestructura prácticamente inusable para los requisitos planteados.

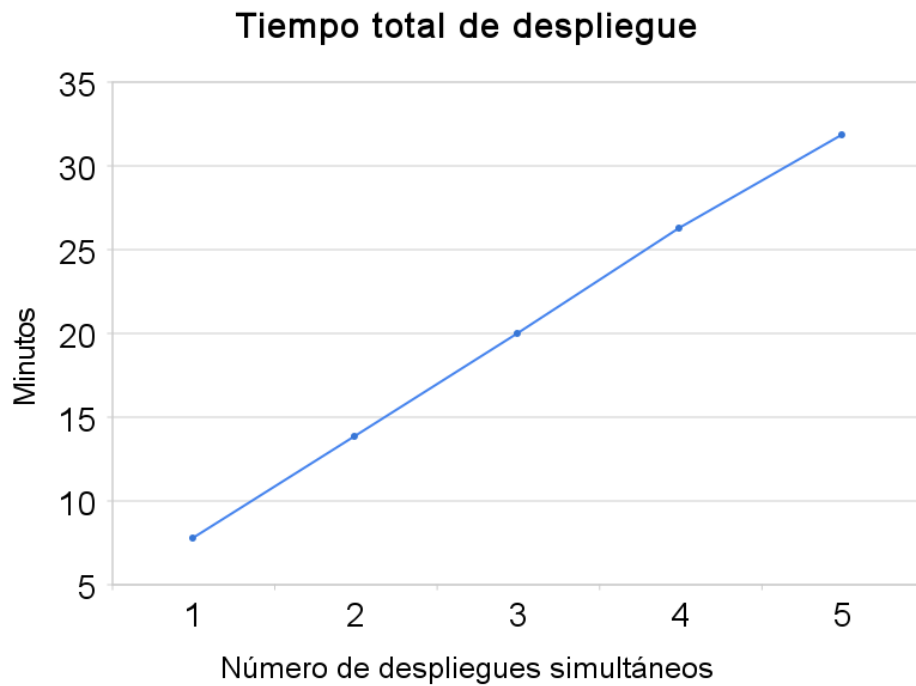


Figura 4.1: Tiempo total de despliegues simultáneos con drivers ssh. Un único host.

Viendo la figura 4.1, podemos sacar conclusiones del rendimiento del sistema, pero no de la causa de ese rendimiento tan bajo. En la figura 4.2 se muestra la mediana del tiempo que necesitan las máquinas virtuales de cada grupo para inicializarse, dividido por tramos. Los datos mostrados son los mismos que los de la figura anterior.

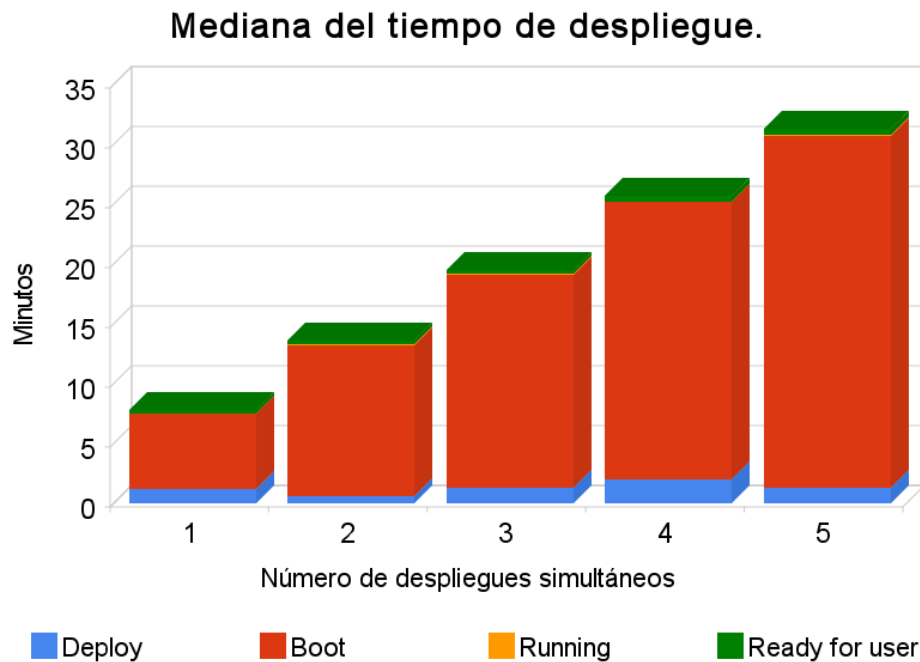


Figura 4.2: Tiempo desglosado de despliegues simultáneos con drivers ssh. Un único host.

Como puede verse, la fase que más tiempo consume del total es la de Boot, en color rojo. La conclusión que puede sacarse es que el número de máquinas virtuales solicitadas simultáneamente influye en el tiempo de despliegue y, además, que la causa principal es el tiempo empleado en realizar las copias mediante scp.

Esta conclusión es totalmente intuitiva y casi podría decirse que obvia si se conoce el funcionamiento interno de OpenNebula. Por lo tanto, sirve para validar el método de medición empleado.

4.2.2. Drivers de transferencia con cache de imágenes

Descripción

Tras comentar la gráfica 4.2, se ha llegado a la conclusión de que la copia de ficheros mediante scp degrada enormemente el rendimiento de la infraestructura. La forma habitual de solucionar este problema sería instalando un sistema de almacenamiento compartido; como NFS, para el cual OpenNebula cuenta con drivers ([28] es un ejemplo de uso).

El nuevo driver de transferencia con cache desarrollado está basado en el driver para transferencias por ssh. La idea es sencilla: si copiar desde el front-end al host tarda demasiado, una copia local debería mejorar el rendimiento considerablemente.

Para desplegar una máquina virtual se necesita hacer una copia de la imagen original. Este nuevo driver asume que la imagen ya se encuentra en el disco duro local del host, por lo que puede realizar una copia mediante el comando `cp` en lugar del anterior `scp`. Con esto no sólo se evita la transferencia por red, también se consigue paralelizar las copias en caso de disponer de varios hosts. De esta manera, desplegar una máquina virtual en un host tardará lo mismo que desplegar cuatro máquinas en cuatro hosts.

Implementación

Copiar el código del driver y explicar con detalle la implementación no aportaría demasiado a lo anteriormente explicado. Pero para dar una idea de lo sencillo que es modificar OpenNebula, este es el cambio más significativo realizado (se omiten otros cambios detalles de menor relevancia):

```
$ diff ssh/tm_clone.sh cached/tm_clone.sh
53c99,100
<     exec_and_log "scp $SRC $DST"
---
>     exec_and_log "ssh $DST_HOST cp $SRC_PATH $DST_PATH"
```

Aunque el driver modificado está escrito en bash, OpenNebula permite usar cualquier lenguaje de programación, ya que los mensajes de control entre el núcleo y los drivers se intercambian por la entrada y salida estándar, usando un protocolo basado en texto. Esta arquitectura modular es uno de los motivos que hacen de OpenNebula una elección adecuada. La arquitectura será mantenible a largo plazo, pudiendo crear nuevos drivers rápidamente para dar soporte a nuevos sistemas de almacenamiento, o nuevos hipervisores que puedan aparecer.

En cuanto a la copia local del repositorio de imágenes que guarda cada host, hay que garantizar que estará actualizada, y que contiene lo mismo en todos los hosts. Rsync [29, 30] es una solución de uso extendido y buen rendimiento.

Rendimiento

En las gráficas del capítulo anterior se vio reflejada la influencia del número de despliegues simultáneos en el tiempo de esos despliegues. Durante la evaluación de este nuevo driver, se han realizado además pruebas con diferentes

números de hosts. De este modo en las siguientes gráficas puede verse la influencia del número de hosts en el tiempo de despliegue, usando el driver con cache del repositorio de imágenes.

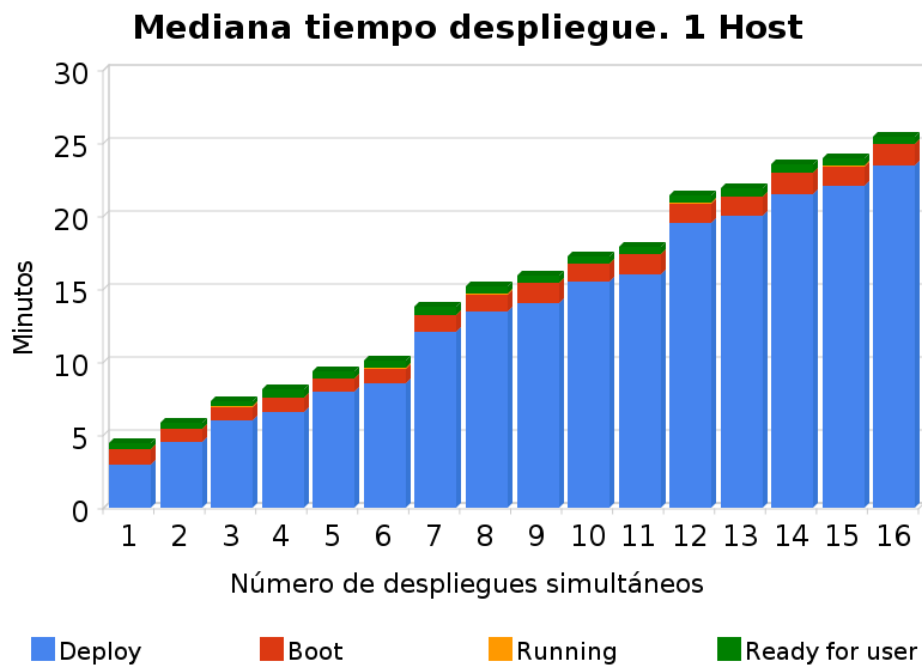


Figura 4.3: Despliegues simultáneos con drivers cache. Un único host.

La figura 4.3 puede compararse a la figura 4.2. Usando los drivers ssh, desplegar un grupo de 5 máquinas virtuales hacía que cada máquina tardase algo más de 30 minutos según la mediana, mientras que con los drivers cache el tiempo es menor a 10 minutos.

Los tiempos totales dan números similares. Para ello, hay que comparar las gráficas 4.1 y 4.7

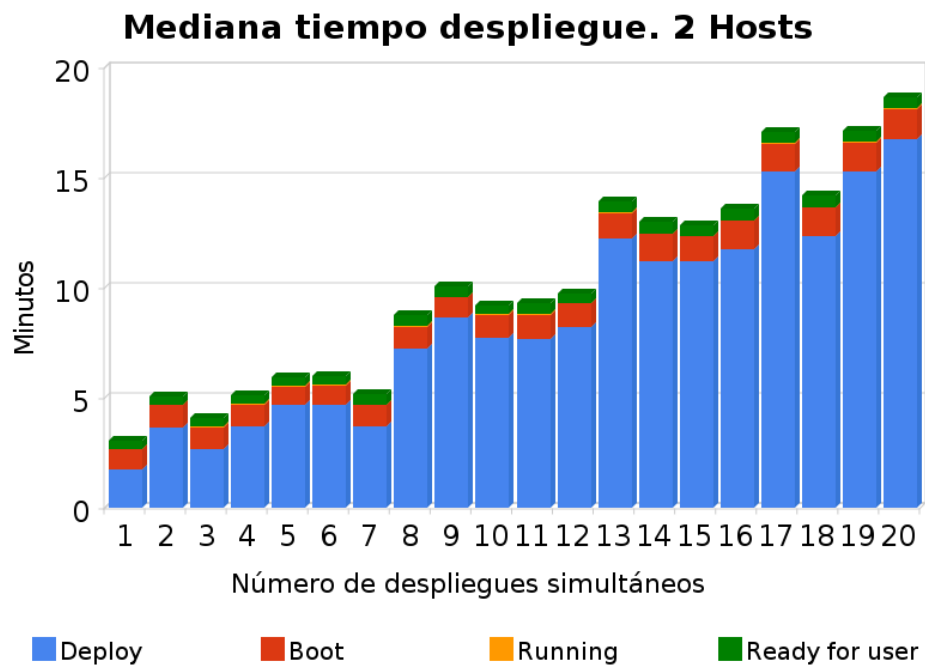


Figura 4.4: Despliegues simultáneos con drivers cache. Dos hosts.

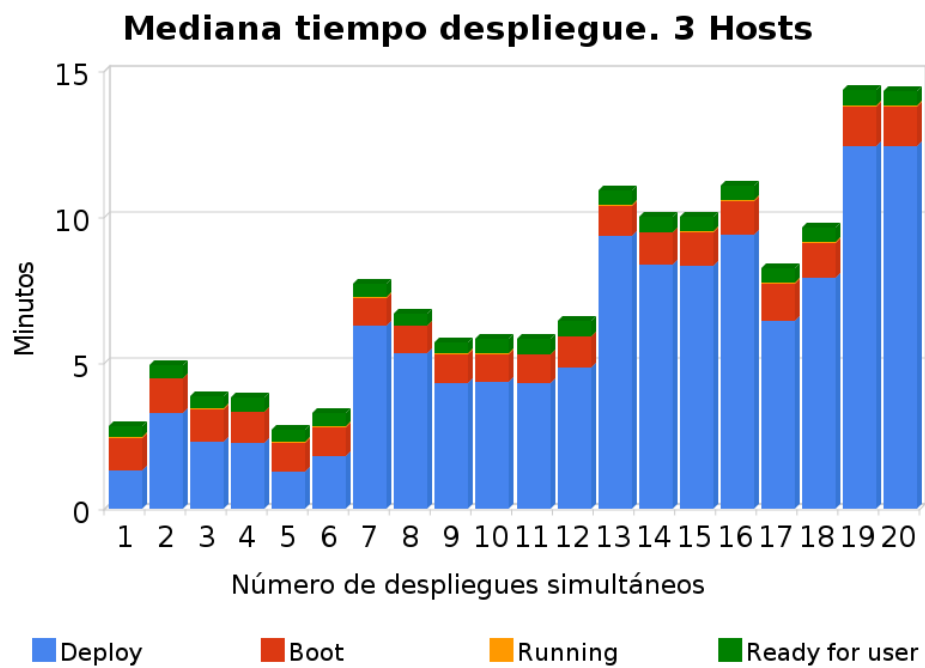


Figura 4.5: Despliegues simultáneos con drivers cache. Tres hosts.

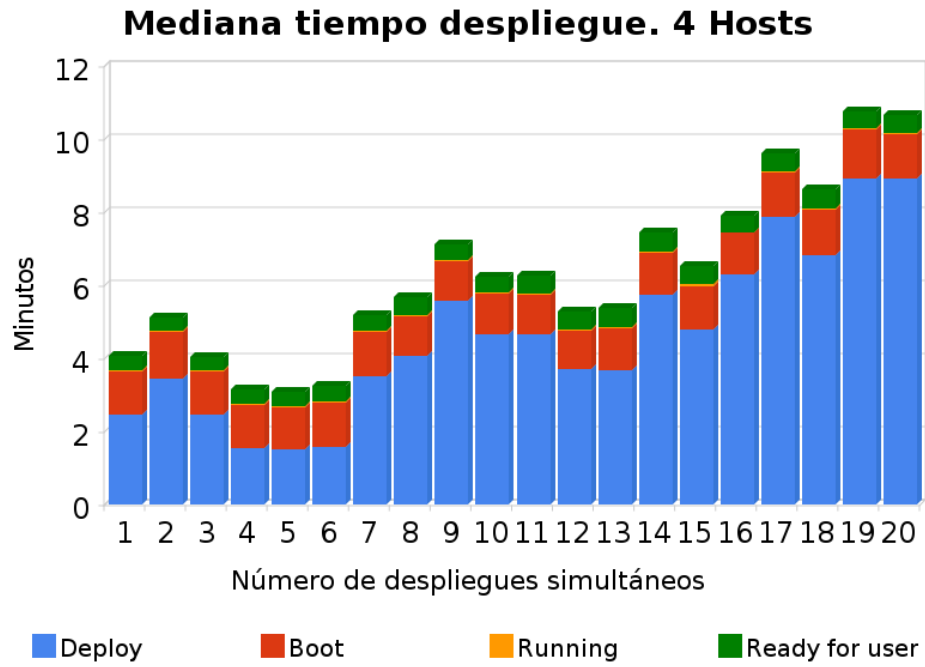


Figura 4.6: Despliegues simultáneos con drivers cache. Cuatro hosts.

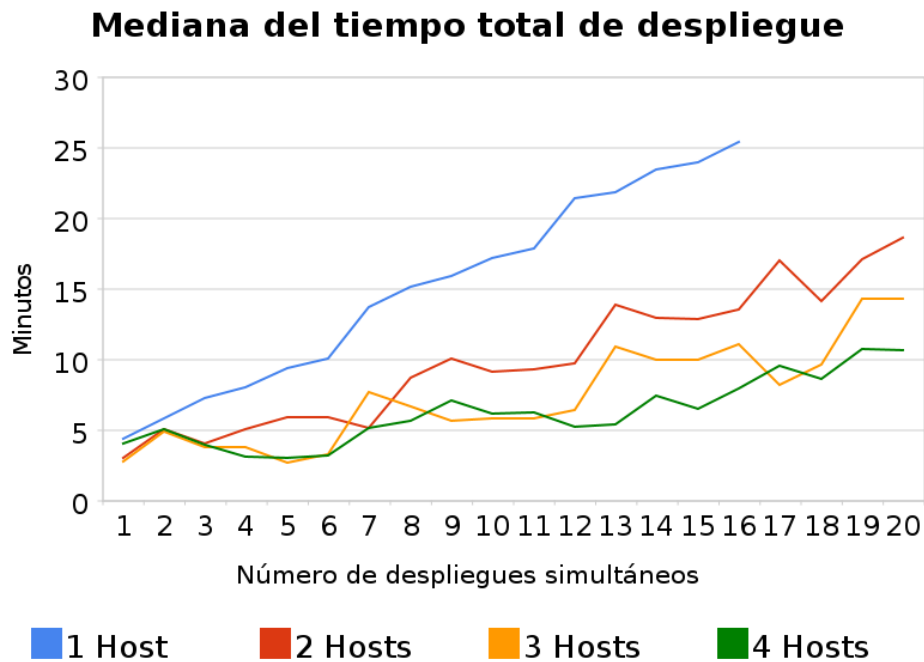


Figura 4.7: Tiempo total de despliegue. Comparativa de cómo influye el número de hosts (hosts).

Las gráficas mostradas en esta sección han demostrado por un lado que el rendimiento del driver de transferencia con cache es mejor que el del driver ssh, y por otro que el número de hosts en uso influye en el tiempo de despliegue de las máquinas virtuales.

Otro resultado que merece la pena comentar es que el tiempo total está claramente definido por el tiempo consumido en la fase boot. Recordemos que el tramo de tiempo representado en azul corresponde al tiempo que transcurre desde que la máquina es solicitada, hasta que el planificador decide crearla en uno de los hosts. Para la realización de este experimento, el planificador fue configurado intentar secuenciar los tiempos de copia de imágenes. De este modo, no se produce coincidencia en las operaciones de copia de ficheros, lo cual tendría un impacto negativo en los tiempos totales de despliegue.

El planificador de OpenNebula por tanto, debe ser ajustado si se quieren obtener los mejores resultados. La modificación de los parámetros más básicos, y la creación de nuevas políticas no es una tarea complicada. Aunque queda fuera de los objetivos de este proyecto explorar políticas de planificación específicas, se puede destacar como referencia para trabajos futuros que existe un planificador mucho más avanzado llamado Haizea [31, 32].

4.2.3. Nuevos drivers con pre-copia

Descripción

Tras comprobar el rendimiento del nuevo driver de transferencia con cache descrito en la sección anterior, se ha llegado a la conclusión de que el tiempo de copia sigue siendo la causa más influyente en el tiempo total de despliegue. Para mejorar por tanto el tiempo total, sería beneficioso seguir bajando ese tiempo de copia, o incluso eliminarlo.

El driver con pre-copia descrito en esta sección se trata de una modificación del driver con cache. La idea principal es realizar las copias de manera anticipada, de manera que cuando se quiera desplegar una máquina virtual se pueda usar sin tener que esperar a realizar la copia.

El repositorio de imágenes, normalmente presenta este aspecto. El directorio contiene dos imágenes registradas:

```
img_repo
|-- c73ad1763babec1eeab7a162cdc32a5e41248cfe
'-- cfa3c1e67234395935d0536d4963175094abd085
```

Para utilizar este nuevo driver, el directorio del repositorio de imágenes (replicado y sincronizado en cada uno de los hosts) se modifica para seguir esta estructura:

```
img_repo
|-- c73ad1763babec1eeab7a162cdc32a5e41248cfe
|-- c73ad1763babec1eeab7a162cdc32a5e41248cfe_copies
|   |-- 01.disk
|   |-- 02.disk
|   |-- 03.disk
|   |-- 04.disk
|   '-- 05.disk
|-- cfa3c1e67234395935d0536d4963175094abd085
'-- cfa3c1e67234395935d0536d4963175094abd085_copies
    |-- 01.disk
    |-- 02.disk
    |-- 03.disk
    |-- 04.disk
    '-- 05.disk
```

Donde los archivos .disk son copias de las imágenes realizadas de antemano.

Cuando se despliega una máquina virtual, se usa una de las copias disponibles. A efectos prácticos, se trata de sustituir el comando cp por el comando mv.

Implementación

Aunque la estructura del directorio con el repositorio de imágenes se ha modificado, los cambios necesarios en el código no suponen un cambio demasiado profundo. El siguiente extracto de código sirve para ilustrar los cambios realizados en el código. Se omiten partes menos relevantes, como el tratamiento de errores e instrucciones que guardan en el log las acciones realizadas.

```
FILE='ssh $DST_HOST "ls -l $SRC_COPIES | head -1"'\nexec_and_log "ssh $DST_HOST mv $SRC_COPIES/$FILE $DST_PATH"
```

El driver no se ocupa de crear las copias en el host, simplemente de usarlas para desplegar las máquinas virtuales. Para crear nuevas copias, la solución más sencilla consiste en utilizar un script ejecutado periódicamente por el demonio de sistema cron.

Existen diferentes problemas con lo que podría encontrarse el driver. A continuación se comentan brevemente estos potenciales problemas, y la forma de solucionarlos.

Usar la misma copia para dos máquinas virtuales

Aunque el movimiento de imágenes necesita muy poco tiempo, podría darse el caso de que el driver intentase desplegar dos máquinas virtuales en el mismo instante. Para evitar que se hagan dos operaciones de mv sobre el mismo archivo, se crea un fichero semáforo que bloquea el uso del directorio.

Esto se implementa con la utilidad de línea de comandos flock, que permite sincronización mediante un fichero. Antes de listar el contenido del repositorio y ejecutar el comando mv, el driver bloquea la carpeta. Por ejemplo, antes de

mover el fichero 01.disk, el directorio tendrá este contenido:

```
img_repo
|-- c73ad1763babec1eeab7a162cdc32a5e41248cfe
|-- c73ad1763babec1eeab7a162cdc32a5e41248cfe_copies
|   |-- .lock
|   |-- 01.disk
|   |-- 02.disk
|   |-- 03.disk
|   |-- 04.disk
|   '-- 05.disk
|-- cfa3c1e67234395935d0536d4963175094abd085
'-- cfa3c1e67234395935d0536d4963175094abd085_copies
    |-- 01.disk
    |-- 02.disk
    |-- 03.disk
    |-- 04.disk
    '-- 05.disk
```

Cualquier acción sobre la carpeta copies tendrá que esperar, mediante el comando flock, a que se libere el archivo .lock.

Usar copias no terminadas

Podría darse el caso de que el driver intente mover una de las copias antes que ésta esté finalizada. Para evitarlo, el script encargado de crear nuevas copias periódicamente las creará del siguiente modo:

Primero, la copia de la imagen se hace con un nombre oculto:

```
img_repo
|-- c73ad1763babec1eeab7a162cdc32a5e41248cfe
|-- c73ad1763babec1eeab7a162cdc32a5e41248cfe_copies
|   |-- .06.disk
|   '-- 05.disk
```

Cuando la copia está terminada, el script bloquea el directorio con el fichero .lock y la utilidad flock. Se cambia entonces el nombre de la copia nueva para

que deje de ser oculta, y se vuelve a liberar el fichero .lock:

```
img_repo
|-- c73ad1763babec1eeab7a162cdc32a5e41248cfe
|-- c73ad1763babec1eeab7a162cdc32a5e41248cfe_copies
|   |-- .lock
|   |-- 05.disk
|   '-- 06.disk
```

Rendimiento

A continuación se muestran las gráficas obtenidas con el nuevo driver con pre-copia. Las pruebas realizadas consisten en desplegar simultáneamente 20 máquinas virtuales, usando los tres drivers disponibles: el driver original ssh, el driver con cache de imágenes, y el driver con pre-copia explicado en esta sección.

Los datos fueron tomados desplegando 20 máquinas virtuales simultáneamente, usando los 4 hosts. Las figuras 4.8 y 4.9; y las 4.11 y 4.11, muestran los mismos datos. Se muestran los drivers con cache y con pre-copia por separado para poder ser comparadas mejor, ya que la diferencia de tiempo entre estos con el driver ssh es mucho más alta que la que existe entre ellos.

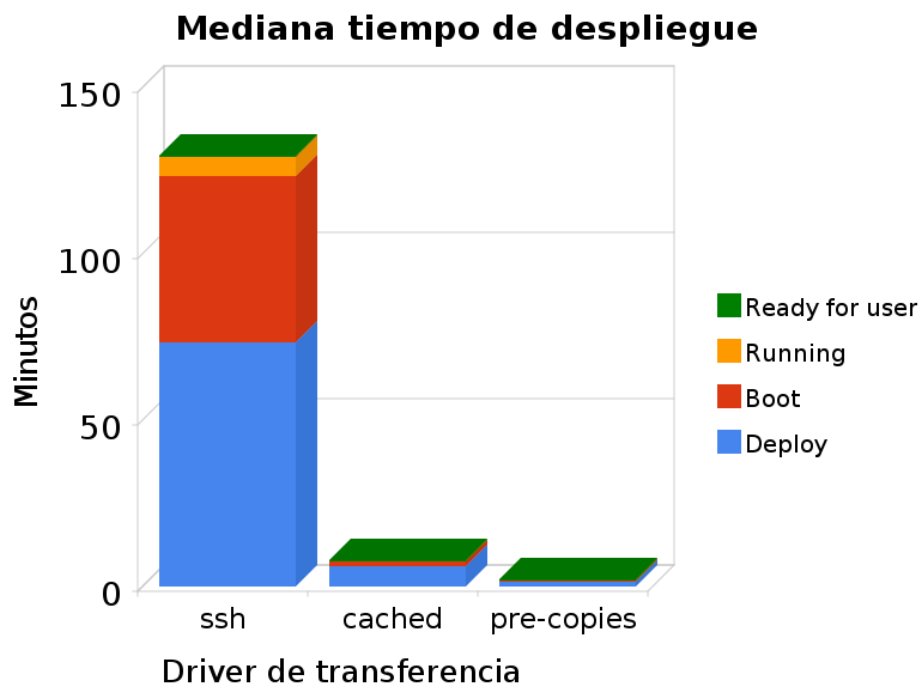


Figura 4.8: Mediana del tiempo necesario para desplegar 20 máquinas virtuales simultáneamente, usando 4 hosts.

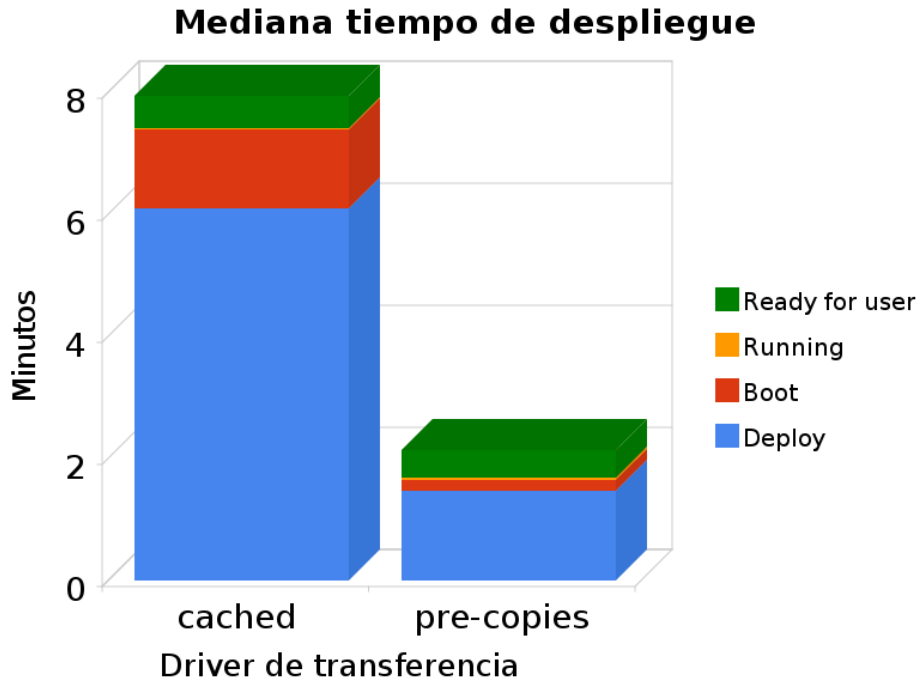


Figura 4.9: Mediana del tiempo necesario para desplegar 20 máquinas virtuales simultáneamente, usando 4 hosts.

Las dos primeras figuras muestran la mejora de rendimiento, mostrando la mediana del tiempo necesario para desplegar 20 máquinas virtuales. La mejora en el rendimiento se ve claramente influenciada por el tiempo de la fase de boot, que hace que se pueda obtener también una fase de deploy más breve.

Las siguientes dos gráficas muestran el tiempo total necesario para desplegar 20 máquinas virtuales simultáneamente. El resultado más destacable es el excepcional rendimiento del driver con pre-copia, que es capaz de lanzar las 20 máquinas en poco menos de 3 minutos, frente a los 14 del driver con cache. El driver original ssh tarda más de 3 horas.

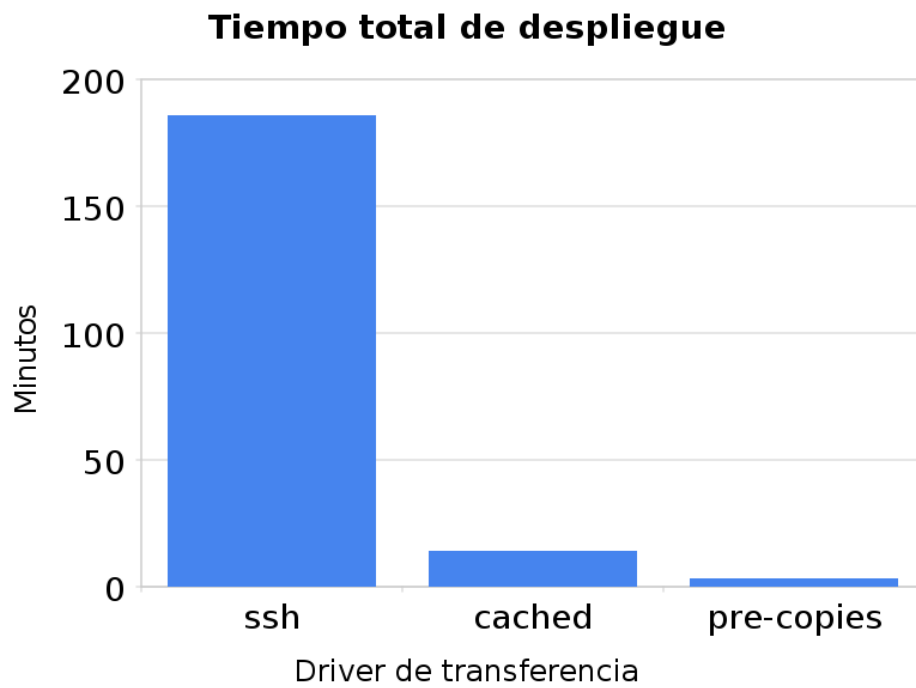


Figura 4.10: Tiempo total necesario para desplegar 20 máquinas virtuales simultáneamente, usando 4 hosts.

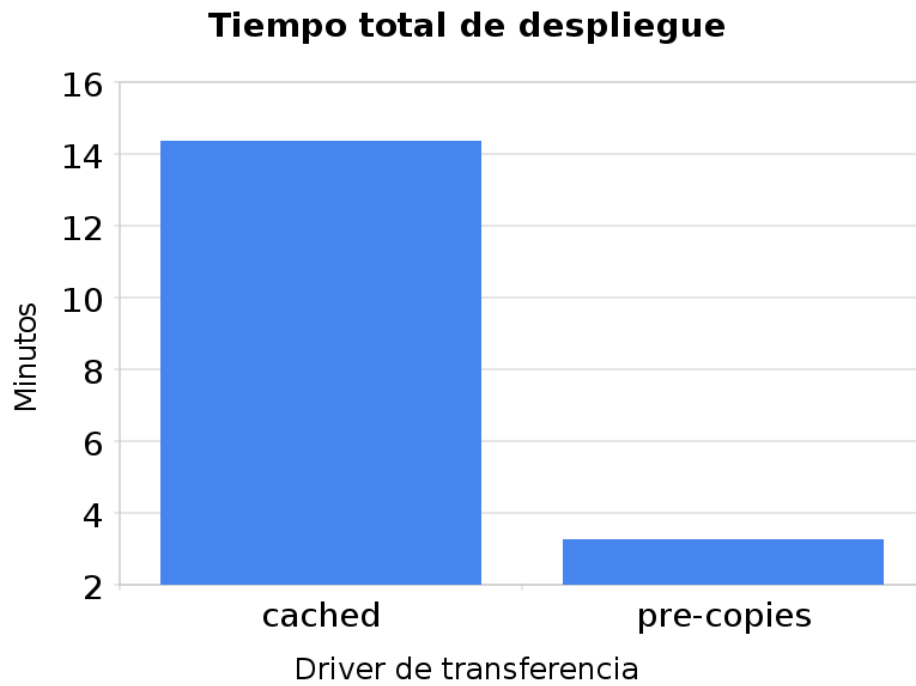


Figura 4.11: Tiempo total necesario para desplegar 20 máquinas virtuales simultáneamente, usando 4 hosts.

4.2.4. Conclusiones

En este capítulo se ha demostrado que con la métrica propuesta, se puede medir el rendimiento de la infraestructura prototipo. Además, se obtiene suficiente grado de detalle como para poder estudiar las causas de un rendimiento bajo.

Por otra parte, con unas mejoras sencillas sobre la instalación base se ha logrado bajar el tiempo de despliegue de 20 máquinas virtuales desde 3 horas a 3 minutos.

En la Facultad, la infraestructura cloud tendrá una mayor carga en momentos previsibles, como son los períodos de exámenes o, concretando aún más, al iniciar las sesiones de laboratorio programadas en el horario semanal. El driver con pre-copia desarrollado podría utilizarse de manera proactiva, creando copias de las máquinas virtuales unos minutos antes del comienzo de la sesión de laboratorio.

Conseguir desplegar 20 máquinas virtuales en 3 minutos puede que no parezca tan relevante cuando se tiene en cuenta el volumen de puestos de trabajo que se manejaría en caso de implantar la infraestructura cloud en la Facultad. Quizás sea más relevante que la infraestructura prototipo haya conseguido tal rendimiento sin contar con ningún sistema de almacenamiento compartido, que sin duda alguna permitiría mejorar los tiempos presentados incluso con una carga de trabajo mayor.

4.3. Medición de la latencia de uso

Como se ha explicado en la sección 4, la latencia de uso es la velocidad de respuesta que el usuario recibe a sus acciones interactivas en el entorno gráfico de las máquinas virtuales. Esta velocidad no tiene que ver con el tiempo que tardan tareas como compilar, procesar datos, cargar una página web, etc. Lo que se busca medir es la velocidad con la que el sistema responde a las acciones del usuario: mover el ratón, escribir texto, abrir un menú desplegable, etc.

Debido a que la aplicación de cloud computing para virtualizar escritorios no está muy extendida, no existen muchos artículos dedicados al rendimiento de este tipo de infraestructuras. Los pocos trabajos existentes sobre el estudio de rendimiento de los escritorios virtualizados en cloud computing coinciden en considerar la agilidad de respuesta como una medida importante del éxito de la infraestructura [22, 23].

4.3.1. VNCPlay

VNCPlay [23] es una herramienta multi-plataforma para medir el rendimiento de aplicaciones gráficas interactivas. VNCPlay graba una sesión interactiva que puede ser reproducida múltiples veces sobre sistemas con diferentes configuraciones, para más tarde comparar el rendimiento obtenido en cada uno. El tiempo de respuesta se calcula comparando los tiempos en los que se ocurren actualizaciones de la pantalla similares en cada una de las sesiones reproducidas.

VNCPlay utiliza el protocolo VNC [33] para grabar y posteriormente reproducir las sesiones. Gracias a esto, puede utilizarse para medir el rendimiento de sistemas Windows o Linux, usando cualquier gestor de ventanas o tema gráfico del mismo.

En el artículo original, VNCPlay es usado para medir la influencia de la velocidad del procesador y la carga de disco sobre las aplicaciones interactivas de escritorio. En este trabajo, se ha experimentado con el uso de VNCPlay para automatizar mediciones de rendimiento de infraestructuras cloud.

Cómo usarlo

El uso básico de VNCPlay es bastante simple, siendo los siguientes pasos el flujo normal de uso:

1. Grabar la sesión.

Usando el siguiente comando, VNCPlay iniciará una sesión VNC con el servidor indicado. La sesión queda guardada en un fichero.

```
$ vncplay record server:port trace.vnc
```

2. Reproducir la sesión en otra máquina virtual.

VNCPlay iniciará una ventana con una sesión VNC, que mostrará cómo se reproduce la sesión automáticamente.

```
$ vncplay play server:port trace.vnc out.rfb
```

3. Analizar las sesiones reproducidas.

Este paso puede llevar casi tanto tiempo como el necesario para reproducir todas las sesiones.

```
$ vncplay analyze out1.rfb ... > analyze.out
```

4. Generar resultados.

Utilizando los archivos intermedios del paso anterior, se generan los datos de forma que puedan plasmarse en gráficas.

```
$ vncanalyze median analyze.out > cdf.out
```

Aunque los pasos anteriores describen todo lo necesario para utilizar la herramienta, existen algunos detalles que deben tenerse en cuenta a la hora de usar VNCPlay.

Con respecto a la grabación de sesiones, uno no puede guardar cualquier tipo de sesión. Por ejemplo, no sería viable grabar a un alumno mientras realiza un sesión de laboratorio práctica real, ya que VNCPlay está orientado al uso mediante ratón.

Para que una sesión pueda ser reproducida fielmente más tarde, debe tenerse en cuenta que los puntos de sincronización son los clicks de ratón. Si se hace click en un menú, por ejemplo Editar, y luego en la opción desplegable Pegar, al reproducir la sesión VNCPlay esperará a que el menú desplegable esté listo antes de hacer click en el sitio que debería ocupar la entrada Pegar. Esta acción podrá reproducirse aunque el sistema operativo tarde minutos en responder al primer click, ya que hasta que VNCPlay no reciba la actualización de la pantalla con la entrada Pegar visible, no realizará el segundo click.

Se deben evitar ciertas acciones, como por ejemplo posar el ratón sobre un menú para desplegar un sub-menú de segundo nivel. Esto puede hacer que al reproducir la sesión, VNCPlay mueva el ratón hacia el espacio que ocupará el sub-menú antes de que éste aparezca, quedando interrumpida la reproducción. VNCPlay en ese caso esperará indefinidamente a la sincronización de pantalla con el sub-menú desplegado.

Merece también ser mencionado el hecho de que VNCPlay requiere un entorno gráfico para poder reproducir las sesiones guardadas, lo cual es un inconveniente si se utiliza para realizar tests automáticos. En la infraestructura prototipo, las sesiones fueron reproducidas desde el front-end, que carece de servidor X.

Para poder reproducir las sesiones sin tener que instalar todo un entorno gráfico, puede utilizarse xvfb (X virtual framebuffer). Xvfb permite que aplicaciones que requieren una interfaz gráfica se ejecuten en un framebuffer virtual. Su página en x.org [34] no es demasiado completa, pero existen en internet tutoriales con ejemplos de uso prácticos.

4.3.2. Rendimiento de la instalación

Lo que se pretende en este experimento es comprobar cómo influye el número de máquinas virtualizadas en un host en su latencia de uso.

Para grabar y reproducir la sesión VNCPlay, se ha virtualizado el sistema operativo Ubuntu 10.04. A esta máquina virtual, ejecutada en uno de los cuatro hosts de la infraestructura prototipo, se accede mediante VNC. El servidor VNC es creado por el hipervisor KVM, si así se solicita a OpenNebula en la plantilla con la que se crea la máquina virtual.

La sesión grabada no es más que un ejemplo de uso aleatorio del sistema. Durante la sesión, se abren carpetas, se navega por ellas, y se usa alguna aplicación como OpenOffice. Podría argumentarse que para ser rigurosos, la sesión debería incluir el uso habitual que se le daría a la máquina virtualizada en una sesión de laboratorio: abrir un entorno de desarrollo, compilar algún trabajo, consultar archivos pdf, navegar por la web, etc. En realidad, esto sería interesante como un trabajo futuro más extenso: estudiar cómo influye el perfil de cada asignatura en el rendimiento de las máquinas virtuales, y cómo garantizar que cada máquina recibe los recursos necesarios para que no se ralentice. En este trabajo no se profundiza tanto en el uso de VNCPlay, y se tiene como objetivo suficiente el poder automatizar el proceso de medición, y validarlo con la siguiente gráfica.

La gráfica 4.12 muestra la mediana del tiempo de respuesta a las acciones interactivas sobre el escritorio virtualizado. El eje horizontal muestra el número de máquinas virtualizadas simultáneamente en el mismo host. Sobre cada máquina virtual, se ha reproducido una sesión VNCPlay, y lo que se muestra es la mediana de los resultados del grupo, no de una única máquina.

Antes de comenzar la reproducción de la sesión VNCPlay, se ha esperado a que todas las máquinas virtuales estuviesen completamente inicializadas, para así garantizar que las sesiones interactivas se realizarían solapándose y no secuencialmente.

Al acabar las sesiones VNCPlay sobre cada grupo de máquinas, éstas han sido apagadas para que el siguiente grupo comenzase desde un mismo estado inicial.

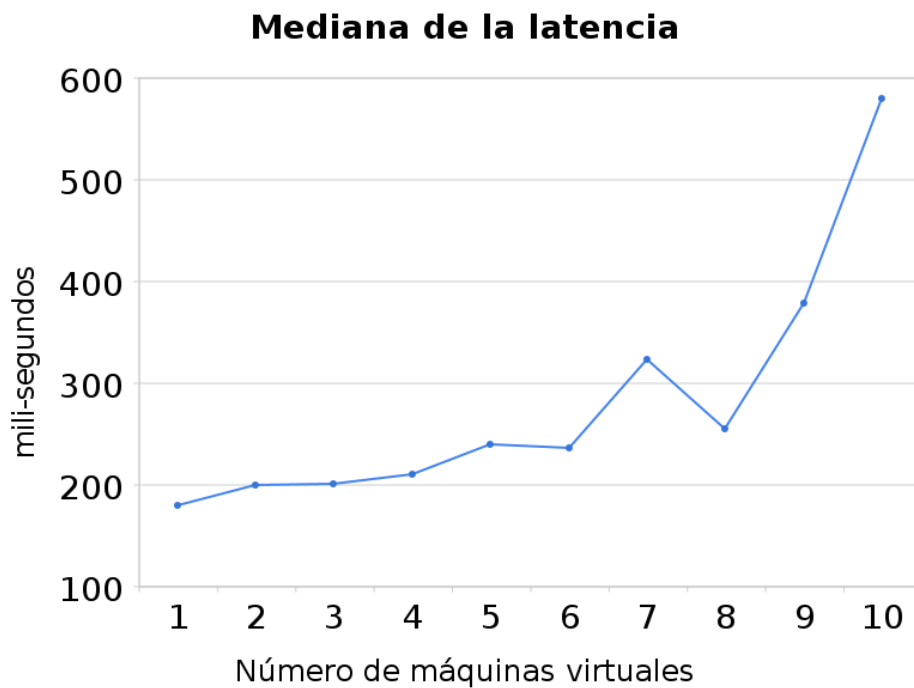


Figura 4.12: Influencia del número de máquinas virtuales en un único host sobre la latencia de uso.

Tras observar algunas de las sesiones reproducidas, se puede afirmar que con un tiempo de respuesta de 500 milisegundos o menor la experiencia de uso es fluída. En la figura 4.12 se ve que aunque el número de máquinas influye en la latencia de la respuesta, no lo hace de manera lineal. A partir de 9 máquinas, la latencia aumenta de manera más abrupta. Esto se debe seguramente a que el host cuenta con 8 núcleos de CPU.

Por lo tanto, podemos concluir que si queremos conseguir una experiencia de uso fluída, para esta máquina virtual en concreto, y con un uso similar al grabado en la sesión VNCPlay, el número de máquinas virtualizadas no debe sobrepasar el número de núcleos de CPU del host.

OpenNebula ya soporta este tipo de restricciones. En este caso, para conseguir que a partir de 8 máquinas virtuales el planificador deje de enviar más máquinas al mismo host, hay que incluir la siguiente línea en la plantilla de la máquina virtual:

```
CPU = 1
```

Eso hará que la máquina solicite una CPU física para uso exclusivo. Si de los datos de las gráficas hubiésemos concluido que no se debería sobrepasar

la relación de 2 máquinas por cada CPU, la plantilla tendría que contener el atributo CPU siguiente:

$$\text{CPU} = 0.5$$

Así, cada máquina reservaría media CPU.

Capítulo 5

Conclusiones y Trabajo Futuro

En este trabajo se ha descrito la posibilidad de aplicar una arquitectura cloud computing para virtualizar puestos de laboratorios, haciendo un estudio de las limitaciones de los laboratorios actuales y cómo un cloud privado ayudaría a solucionarlos. El escenario tomado como referencia a la hora de estudiar los requisitos ha sido la Facultad de Informática de la Universidad Complutense de Madrid, ya que se considera que podría beneficiarse de la implantación de una infraestructura de estas características.

Para avanzar en el estudio de esta propuesta, se ha diseñado una arquitectura cloud. Este diseño no sólo se ha descrito teóricamente, sino que se ha creado un prototipo funcional con el fin de validarlo. En la memoria se describe la creación de una infraestructura cloud privada que permite virtualizar entornos de escritorio.

Se ha realizado además una propuesta de métricas que definen el rendimiento de una infraestructura desktop cloud para virtualizar puestos de laboratorio. Estas métricas se han probado con la infraestructura prototipo, desarrollando scripts que automatizan las mediciones y la generación de datos resultantes.

Los resultados de aplicar los tests de rendimiento desarrollados sobre la infraestructura prototipo avalan su viabilidad: la arquitectura planteada podría llevarse a una infraestructura física de mayor escala, y comenzar a usarse como programa piloto en alguna de las asignaturas impartidas.

También ha quedado probada la capacidad de escalar el cloud. Con tan sólo activar o desactivar hosts físicos de la arquitectura, podemos dedicar más recursos al cloud sin necesidad de realizar ajustes de ningún tipo. Esta escalabilidad será clave si algún día se implanta esta solución cloud en la Facultad, ya que frente a un aumento en la demanda de uso la infraestructura puede crecer fácilmente adquiriendo nuevo hardware; o incluso crecer temporalmente contratando capacidad a proveedores cloud externos como Amazon EC2.

La infraestructura diseñada, aunque totalmente funcional como cloud, no aspira ser una solución completa que pueda comenzar a utilizarse por alumnos y profesores. Esta memoria constituye el primer paso en la línea de investigación a seguir durante el doctorado. Durante la realización de la memoria, se han identificado las siguientes líneas de investigación para el trabajo futuro.

Con respecto a la medida del tiempo de despliegue, sería interesante aprovechar la información recogida no sólo para medir el rendimiento del sistema, sino también para desarrollar nuevas políticas de planificación. Unas políticas de planificación inteligentes y proactivas, adaptadas a los requisitos específicos de laboratorios virtualizados y el modo en que se usan, supondrían una aportación al nuevo paradigma desktop cloud.

Por otro lado, la caracterización de cargas de trabajo en entornos interactivos sería una forma de especializar los entornos cloud genéricos al desktop cloud. Estos estudios de carga de uso serían valiosos para poder crear clouds que garanticen una calidad de servicio (QoS, Quality of Service).

Bibliografía

- [1] D. Hardaway, M. Hogan, and R. Mathieu, "Outsourcing the university computer lab," *Computer*, vol. 38, no. 9, pp. 100–102, 2005.
- [2] N. gentschen Felde, T. Lindinger, and H. Reiser, "Virtualizing an IT Lab for Higher Education Teaching," *1. GI/ITG KuVS Fachgespräch „Virtualisierung“*, p. 91.
- [3] V. Powell, J. Turcheck, P. Wu, L. Franzi, R. Johnson, I. Parker, *et al.*, "VLabNet: A Virtual Laboratory Environment for Teaching Networking and Data Communications," *ISECON*, 2007.
- [4] B. Anderson and T. Daniels, "Xen worlds: Xen and the art of computer engineering education," in *Proceedings of 2006 ASEE Annual Conference and Exposition*, 2006.
- [5] H. Bulbrook, "Using Virtual Machines to provide a secure Teaching Lab environment," *White Paper, Durham technical College, Durham, USA*.
- [6] www.amazon.com/ec2.
- [7] A. Kochut, K. Beaty, H. Shaikh, and D. Shea, "Desktop workload study with implications for desktop cloud resource optimization," in *Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, pp. 1–8, IEEE, 2010.
- [8] K. Miller and M. Pegah, "Virtualization: virtually at the desktop," in *Proceedings of the 35th annual ACM SIGUCCS conference on User services*, p. 260, ACM, 2007.
- [9] P. Xu, X. Huang, Y. Wu, L. Liu, and W. Zheng, "Campus Cloud for Data Storage and Sharing," in *2009 Eighth International Conference on Grid and Cooperative Computing*, pp. 244–249, IEEE, 2009.
- [10] R. Creasy, "The origin of the VM/370 time-sharing system," *IBM Journal of Research and Development*, vol. 25, no. 5, pp. 483–490, 1981.

- [11] R. Goldberg, "Survey of virtual machine research," *IEEE Computer*, vol. 7, no. 6, pp. 34–45, 1974.
- [12] J. Tate, K. Gebuhr, K. Jayaraman, and X. Zhao, "IBM System Storage SAN Volume Controller,"
- [13] T. Petrović and K. Fertalj, "Demystifying desktop virtualization," in *9th WSEAS International Conference on Applied Computer Science (ACS'09)*, 2009.
- [14] www.microsoft.com/windows/virtual-pc/.
- [15] www.vmware.com/products/workstation/.
- [16] www.qemu.org.
- [17] www.vmware.com/esx.
- [18] www.virtualbox.org.
- [19] www.linux-kvm.org.
- [20] www.xen.org.
- [21] A. Theurer, K. Rister, and S. Dobbelsstein, "A Survey of Virtualization Workloads," in *Proc. of the 2008 Linux Symposium*, vol. 2, pp. 215–225, 2008.
- [22] J. Rhee, A. Kochut, and K. Beaty, "DeskBench: flexible virtual desktop benchmarking toolkit," in *Proceedings of the 11th IFIP/IEEE international conference on Symposium on Integrated Network Management*, pp. 622–629, Institute of Electrical and Electronics Engineers Inc., The, 2009.
- [23] N. Zeldovich and R. Chandra, "Interactive performance measurement with VNCplay," in *Proceedings of the annual conference on USENIX Annual Technical Conference*, p. 54, USENIX Association, 2005.
- [24] B. Quetier and F. Cappello, "A survey of Grid research tools: simulators, emulators and real life platforms," in *17th IMACS World Congress*, 2005.
- [25] X. Liao, H. Jin, L. Hu, and H. Liu, "Towards virtualized desktop environment," *Concurrency and Computation: Practice and Experience*, vol. 22, no. 4, pp. 419–440, 2010.
- [26] M. Abaza, "The Effect of Machine Virtualization on the Environmental Impact of Desktop Environments," 2009.
- [27] K. Beaty, A. Kochut, and H. Shaikh, "Desktop to cloud transformation planning," 2009.

- [28] L. Wang, J. Tao, M. Kunze, A. Castellanos, D. Kramer, and W. Karl, "Scientific cloud computing: Early definition and experience," in *10th IEEE International Conference on High Performance Computing and Communications, 2008. HPCC'08*, pp. 825–830, 2008.
- [29] A. Tridgell, "Efficient algorithms for sorting and synchronization," *Doktorarbeit, Australian National University*, 1999.
- [30] rsync.samba.org.
- [31] B. Sotomayor, R. Montero, I. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," *IEEE Internet Computing*, vol. 13, no. 5, pp. 14–22, 2009.
- [32] haizea.cs.uchicago.edu.
- [33] T. Richardson, Q. Stafford-Fraser, K. Wood, and A. Hopper, "Virtual network computing," *IEEE Internet Computing*, vol. 2, no. 1, pp. 33–38, 1998.
- [34] www.x.org/archive/X11R6.8.2/doc/Xvfb.1.html.